

Antti Jaakkola

## **Dynaaminen reititys erityisvälitysverkoissa**

**Sähkötekniikan korkeakoulu**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi  
diplomi-insinöörin tutkintoa varten Espoossa 19.05.2015.

**Työn valvoja:**

Prof. Jukka Manner

**Työn ohjaaja:**

DI Risto Järvinen

Tekijä: Antti Jaakkola

Työn nimi: Dynaaminen reititys erityisvälitysverkoissa

Päivämäärä: 19.05.2015

Kieli: Suomi

Sivumäärä: 7+59

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka

Koodi: S-38

Valvoja: Prof. Jukka Manner

Ohjaaja: DI Risto Järvinen

Erityisvälitysverkot ovat tietoverkkoja, joissa normaalien tietoverkkojen olemukset pienistä viiveistä, olemattomista pakettihäviöistä ja päästä-päähän-yhteydellisyydestä eivät toteudu. Tässä työssä esitellään ja toteutetaan dynaaminen reititys olemassa olevalle erityisvälitysverkossa toimivalle Multi Interface Communication Software (MICS) -sanomanvälitysjärjestelmälle.

MICS on useasta komponentista koostuva haasteellisten verkkojen sanomanvälitysjärjestelmä, jonka osat tarjoavat reitityksen kannalta hyödyllisiä ominaisuuksia reitityskomponentille. MICS-verkkojen rakenteessa erityisen hitaat ja epävakaat tiedonsiirtolinkit sijaisevat tyypillisesti verkon reunalla. Erityisesti näitä ominaisuuksia ja verkon hierarkista rakennetta hyödyntämällä työssä toteutettu reitityssovellus saadaan toimimaan haasteellisessa ympäristössä toivotulla tavalla.

Työssä suoritettujen testien perusteella voidaan sanoa, että toteutettu reitityssovellus vastaa tavoitteita ja toimii vaaditulla tavalla testiympäristössä, mutta toimivuuden varmistamiseksi käytännön verkoissa vielä kattavamman testauksen suorittaminen on suositeltavaa.

Avainsanat: Reititys, IS-IS, DTN, MICS, DTLSR, MANET, ICMAN

Author: Antti Jaakkola

Title: Dynamic Routing in Challenged Networks

Date: 19.05.2015

Language: Finnish

Number of pages:7+59

Department of Communications and Networking

Professorship: Networking technology

Code: S-38

Supervisor: Prof. Jukka Manner

Advisor: M.Sc. (Tech.) Risto Järvinen

In the challenged networks standard assumptions of low latency, end-to-end connectivity and high probability of packets being received will not be fulfilled. In this thesis a dynamic routing solution for Multi Interface Communication Software (MICS), which is messaging system for the challenged networks, is presented and implemented.

MICS is a modular messaging system for challenged networks, that offers beneficial information for the routing component. In MICS networks slow and unstable connections are located typically at the edge of the network. By using particularly these properties and hierarchical nature of the used network, the implemented routing solution will be able to operate in challenged networks in a desirable way.

Based on the test results, the implemented routing solution fulfills the requirements and operates successfully in the test network. However, in order to ensure correct operation in the practical networks, more comprehensive testing is recommended.

Keywords: Routing, IS-IS, DTN, MICS, DTLSR, MANET, ICMAN

## Esipuhe

Haluan kiittää erityisesti työni valvojaa, professori Jukka Manneria tämän mielenkiintoisen diplomityöaiheen mahdollistamisesta sekä työn aikana saadusta tuesta ja palautteesta. Lisäksi Risto Järvisen väsymätön ja kannustava asenne työni ohjaamiseen ja palautteen antamiseen oli ensiarvoisen tärkeää työn toteutumisen kannalta. Kiitos.

Kiitos lisäksi kaikille entisille työkavereilleni ja nykyisille ystävilleni joihin tutustuin työskennellessäni 5,5 vuoden ajan Aalto-yliopiston Tietoliikenne- ja tietoverkkotekniikan laitoksella. Teiltä aina tarvittaessa saamani apu on ollut korvaamatonta.

Haluan myös kiittää koko perhettäni, mutta erityisesti isääni Heikkiä kaikesta elämäni aikana saamastani tuesta. Ilman teitä tämä työ ei olisi koskaan valmistunut.

Otaniemi, 10.05.2015

Antti T. Jaakkola

# Sisältö

<b>Tiivistelmä</b>	<b>ii</b>
<b>Tiivistelmä (englanniksi)</b>	<b>iii</b>
<b>Esipuhe</b>	<b>iv</b>
<b>Sisällysluettelo</b>	<b>v</b>
<b>Lyhenteet</b>	<b>vi</b>
<b>1 Johdanto</b>	<b>1</b>
<b>2 Reititys tietoliikenneverkoissa</b>	<b>4</b>
2.1 Johdatus reititykseen . . . . .	4
2.2 Graafiteoria . . . . .	5
2.3 Etäisyysvektoriprotokollat . . . . .	8
2.4 Linkkitilaprotokollat . . . . .	10
2.5 Reititys internetissä . . . . .	12
2.6 Erityisvälitysverkot . . . . .	13
2.7 Erityisvälitysverkkojen reititysprotokollia . . . . .	17
2.8 Yhteenveto . . . . .	20
<b>3 Dynaaminen reititys MICS-ympäristössä</b>	<b>22</b>
3.1 Multi Interface Communication Software . . . . .	22
3.2 MICS-verkkojen topologiat . . . . .	29
3.3 Käytetty linkkitilareititysprotokolla . . . . .	30
3.4 Dynaaminen reititys MICS-verkossa . . . . .	34
3.5 Vertailu DTLSR-prokollan kanssa . . . . .	41
3.6 Yhteenveto . . . . .	42
<b>4 Toteutus</b>	<b>44</b>
4.1 Sovellusarkkitehtuuri . . . . .	44
4.2 MICS-spesifiset lisäykset . . . . .	45
4.3 Testaus . . . . .	48
4.4 Kehittämiskohteet . . . . .	51
4.5 Yhteenveto . . . . .	53
<b>5 Yhteenveto</b>	<b>54</b>
<b>Viitteet</b>	<b>56</b>

## Lyhenteet

AS	Autonomous System eli Autonominen järjestelmä, reititettävä yhden tahon hallinnoima osa internetiä.
AODV	Ad Hoc On-Demand Distance Vector, MANET-verkoissa käytetty reititysprotokolla
BFD	Bidirectional Forwarding Detection, virheenhavaitsemisprotokolla.
BGP	Border Gateway Protocol, reititysprotokolla.
CIDR	Classless Inter-Domain Routing, luokaton reititys. Tapa jakaa IP-osoitteita verkkoihin.
CSNP	Complete Sequence Number PDU, täydellinen sekvenssinumeroviesti.
DTLSR	Delay Tolerant Link-State Routing, linkkitilapohjainen reititysprotokolla DTN-verkkoon.
DTN	Delay-Tolerant Networking, viive- ja häiriösietoiset tietoverkot.
ECDN	Eventually Connected Dynamic Network
ERDN	Eventually Routable Dynamic Network
ETDN	Eventually Transportable Dynamic Network
GUI	Graphical User Interface, graafinen käyttöliittymä.
HF	High Frequency, radiotaajuusalue (3-30 MHz).
IARP	Intrazone Routing Protocol, ZRP-hybridireititysprotokollan käyttämä osaprotokolla.
ICMAN	Intermittently Connected Mobile Ad Hoc Network, MANET- ja DTN-verkkojen yhdistelmäverkko.
IERP	Interzone Routing Protocol, ZRP-hybridireititysprotokollan käyttämä osaprotokolla.
IGP	Interior Gateway Protocol, autonomisen järjestelmän sisäinen reititysprotokolla.
IP	Internet Protocol, internetin yhtenäistävä pakettiprotokolla.
IPC	Inter-Process Communication, prosessien välinen viestintä.
IPN	InterPlanetary Networking, planeettojen välinen viestintä.
IS-IS	Intermediate System to Intermediate System, reititysprotokolla.
LSP	Link-State PDU, IS-IS-protokollan linkkitilan välitykseen käyttämä viestityyppi.
LXC	Linux Container, erittäin kevyt virtualisointiympäristö Linuxille.
MANET	Mobile Ad Hoc Network, itsenäisten liikkuvien jäsenien muodostama langaton verkko.
MICS	Multi Interface Communication Software, haasteellisten verkkojen sanomanvälitysjärjestelmä.
MPLS	Multiprotocol Label Switching, tunnelointiprotokolla.
MPLSR	Multi-Policy Link State Routing, DTN-verkkojen reititysprotokolla.
NLng	NetLink next generation, laajennus Netlink-protokollaan.

OLSR	Optimized Link State Routing, MANET-verkoissa käytetty reititysprotokolla.
OSPF	Open Shortest Path First, reititysprotokolla.
PDU	Protocol Data Unit, IS-IS-protokollan lähettämä yksittäinen viesti.
PSNP	Partial Sequence Number PDU, osittainen sekvenssinumeroviesti.
RIP	Routing Information Protocol, reititysprotokolla.
ROCO	ROuting COre, MICS-järjestelmän käyttämä Reititysydin.
SCF	Store-Carry-Forward, DTN-verkoissa käytetty reititysmenetelmä.
SMS	Short Message Service, tekstiviesti.
SPF	Shortest Path First, tietoverkoissa usein käytetty reitinlaskenta-algoritmi.
TCP	Transmission Control Protocol, siirtokerroksella toimiva luotettava tiedonsiirtoprotokolla.
TLV	Type-Length-Value, tiedonsiirtoprotokollien käyttämä tyyppin ja pituuden sisältävä koodaustapa valinnaiselle tiedolle.
VHF	Very High Frequency, radiotaajuusalue (30-300 MHz).
ZRP	Zone Routing Protocol, hybridireititysprotokolla.

# 1 Johdanto

Maailman suurin tietoliikenneverkko internet on suunniteltu olettaen, että verkko on yhtenäinen, viiveet suhteellisen pieniä ja pakettihäviöt ensisijaisesti merkki verkon ruuhkautumisesta. Näiden oletuksien pohjalta on rakennettu monista autonomisista verkoista hyvin toimiva kokonaisuus, jota nykyään käyttää jo lähes puolet maapallon väestöstä.

Kaikki käytössä olevat tietoliikenneverkot eivät kuitenkaan pysty toteuttamaan näitä oletuksia. Planeettojen välisessä viestinnässä etäisyydet viestivien osapuolien välillä ovat niin pitkiä, että viiveet kasvavat valtavan suuriksi. Kehittyvällä alueella sijaitseva kylä voi olla yhteydessä muuhun maailmaan pari kertaa viikossa saapuvan linja-auton toimiessa välittäjänä, jolloin suurimman osan ajasta kylä on erillään muusta verkosta. Katastrofi- tai sotilasalueilla verkoissa voi olla paljon häiriöitä, josta syystä pakettihäviöt ovat ajoittain tai jatkuvasti erittäin suuria. Lisäksi esimerkiksi sensoriverkoissa sekä verkon siirtokapasiteetti että solmujen resurssit voivat olla erittäin pieniä. Nämä ovat muutamia esimerkkejä tilanteista, joissa internetin rungon muodostavat protokollat eivät toimi halutulla tavalla. Näitä haasteellisia verkkoja kutsutaan erityisvälitysverkoiksi.

Reititys on tietoliikenneverkkojen perusominaisuus, joka pyrkii selvittämään optimaalisen tavan välittää viestejä verkossa solmujen välillä. Reitityksen kannalta erityisvälitysverkot asettavat mielenkiintoisen haasteen: viestien välityksen mahdollistaminen dynaamisesti muuttuvassa heterogeenisissä verkossa mahdollisimman tehokkaasti ilman, että reitityksen kontrolliliikenne tukkii hitaita linkkejä ja näin estää hyötykuorman lähettämisen verkossa.

Haasteista huolimatta erityisvälitysverkkojen reititys voidaan hoitaa usealla eri tavalla riippuen käytössä olevan verkon ominaisuuksista. Usein käytetään hyväksi solmujen talleta-kanna-välitä -toiminnallisuutta. Täysin autonomisissa verkoissa, joista puuttuu hierarkia eikä kaikki solmut ole toistensa tavoitettavissa, käytetään yleisesti tulvittavia reititysratkaisuja. Tulvituksessa sanoma välitetään eteenpäin verkossa mahdollisesti jopa kaikille vatsaan tuleville solmuille, jotka tallettavat sanoman muistiinsa ja välittävät sitä eteenpäin olettaen, että sanoma joskus saapuu perille vastaanottajalle. Tämä lähestystapa on helppo toteuttaa, mutta kuormittaa verkkoa usein tarpeettoman paljon. Tulvittamista voidaan myös optimoida tallentamalla verkon historiatietoja ja laskemalla sen pohjalta todennäköisyyksiä verkon tulevasta rakenteesta. Jos verkon voidaan olettaa pysyvän yhtenäisenä tai verkolla on tunnettua hierarkiaa, voidaan monessa tapauksessa hyödyntää internetissä käytettyjä reititysprotokollia ja muokata niistä käytettävään verkkoon sopivia versioita.



## Työn tavoitteet ja rakenne

Erityisvälitysverkot ovat laaja käsite, ja niiden monimuotoisuudesta johtuen täysin yleiskäyttöistä reititysprotokollaa erityisvälitysverkoille ei ole mahdollista toteuttaa. Tässä työssä keskitytään toteuttamaan reititysmalli haasteelliselle mutta hierarkiselle erityisvälitysverkolle, jossa heikot tietoliikenneyhteydet sijaitsevat erityisesti verkon reunalla. Toteutukselle asetetaan kriteereiksi luotettavuus, skaalautuvuus suuriin verkkoihin, tehokkuus laskennallisessa mielessä, kaistankäytön optimointi ja helppo laajennettavuus ja muokattavuus tulevaisuudessa. Työssä pyritään hyödyntämään sopivassa suhteessa nykyisiä toimiviksi todettuja protokollia ja uusia ideoita optimaallisen ratkaisun löytämiseksi. Tässä luvussa eritellään edellä luoteltuja ominaisuuksia ja määritellään niiden tarkoitukset.

Toteutuksen pitää olla riittävän skaalautuva, jotta reititys jopa tuhansien solmujen verkossa onnistuu. Yleisiä tapoja lisätä reitityksen skaalautuvuutta on jakaa verkko pieniin reititysalueisiin (engl. routing domain) ja käyttää aliverkotusta. Skaalautuvuuden maksimoimiseksi täytyy protokollasuunnittelun ja toteutuksen lisäksi painottaa verkkosuunnittelun ja -parametrien tärkeyttä: väärin konfiguroitu verkko voi rajoittaa mahdollista toimintaa huomattavan paljon, vaikka itse protokolla mahdollistaisi paljon laaja-alaisemman toiminnan.

Vaikka normaaleissa tietoverkoissa reitittimillä on suuri laskentakapasiteetti, erityisvälitysverkoissa reititystä voidaan ajaa resursseiltaan erittäin rajatuissa sulautetuissa ympäristöissä. Tästä johtuen keskeisten tietorakenteiden ja algoritmien valintaan tulee käyttää erityistä huolellisuutta. Lisäksi kontrolliliikenteen minimoiminen on erityisen tärkeää ympäristössä, jossa nopeasti liikenteestä tukkeutuvat hitaat yhteydet ovat osa normaalia verkon rakennetta.

Koska erityisvälitysverkkojen määritelmä on hyvin laaja-ajainen, voi toteutukselle olla tulevaisuudessa käyttöä ympäristöissä mitä ei tätä työtä tehdessä osattu ottaa huomioon. Tästä johtuen toteutuksen pitää olla helposti laajennettavissa ja muokattavissa haluttuihin ympäristöihin ja käyttöskenaarioihin. Suurin tähän vaikuttava tekijä on ohjelmistoarkkitehtuurin huolellinen suunnittelu alusta asti.

Tärkeimmät työssä toteutetut reitityksen optimoinnit olivat käytetyn linkkitila-protokollan laajentaminen tukemaan useaa reititystasoa, omanlaisia reititysalueita ja ilmoittautumista. Reititystasoilla rajoitetaan linkkitilatiedon levittämistä verkossa siten, että linkkitilasanomia levitetään verkossa vain samalla tai ylemmällä reititystasolla oleville naapureille. Poikkeuksena tähän on alin taso 0, joka ei milloinkaan lähetä tai vastaanota reitityssanomiamia. Lisäksi reititystason 0 naapurit mainostetaan linkkitilasanomissa verkkoon vasta kun kyseinen naapuri on suorittanut manuaalisen ilmoittautumisen solmulle. Reititysalueita käytetään myös linkkitilasanomien le-

vityksen rajoittamiseen. Linkkitilasanomia ei tulviteta reititysalueiden välillä, vaan alueiden rajoilla toimivat solmut lähettävät säännöllisin väliajoin tiedon kaikista solmuista, jotka ovat saavutettavissa kauttaan. Tämä vähentää huomattavasti reitityksen aiheuttamaa kontrolliliikennettä verkossa.

Suoritettujen testien perusteella voidaan sanoa, että työssä määritelty ja toteutettu reititys haasteellisten verkkojen sanomanvälitysjärjestelmälle vastaa asetettuja tavoitteita. Suoritetuissa testeissä ei havaittu virheellistä toimintaa, ja käyttämällä työssä toteutettuja optimointia saatiin verkossa lähetettyjen kontrolliviestien määrä pysymään kohtuullisella tasolla. Lisäksi hidasta reunaverkkoa ei toteutetulla ratkaisulla kuormiteta lainkaan. Tästä huolimatta oikean toiminnallisuuden varmistamiseksi käytännön verkoissa on kattavamman testauksen tekeminen suositeltavaa.

Työn rakenne on seuraava. Toisessa luvussa kerrotaan teoreettisella tasolla reitityksestä ja erillisistä käytössä olevista algoritmeista, ja käydään läpi reitityksen erityispiirteitä ja haasteita erityisvälitysverkoissa. Kolmannessa luvussa esitellään ympäristö mihin dynaaminen reititys toteutetaan, tutustutaan tarkemmin yhteen yleisesti käytössä olevaan linkkitilaprotokollaan jonka ympärille totetus rakennetaan, ja käydään läpi protokollaan tehtävät laajennukset ja oletukset reititettävästä verkosta. Neljännessä luvussa esitellään toteutus yleiskäyttöisestä reititysytimeistä, jonka toimintaa on muutettu sopivammaksi haasteellisiin verkkoihin, ja lopuksi summaataan yhteen havainnot ja analysoidaan reititysytimen käytössä saatuja testituloksia.

## 2 Reititys tietoliikenneverkoissa

Tässä luvussa tutustutaan aluksi tietoverkkojen reititykseen korkealla tasolla, johon liittyen käydään läpi olennaisia graafiteorian peruskäsitteitä. Perusteiden jälkeen tutustutaan kahteen reitityksen kannalta hyvin olennaiseen lyhimmän polun ongelman ratkaisevaan algoritmiin ja näiden käytännön sovelluksiin etäisyysvektori- ja linkkitilaprotokollien muodossa. Internetin reitityksen jälkeen siirrytään erityisvälitysverkkojen esittelyyn, josta päästään tutustumaan erityisvälitysverkoissa käytettäviin reititysprotokolliin.

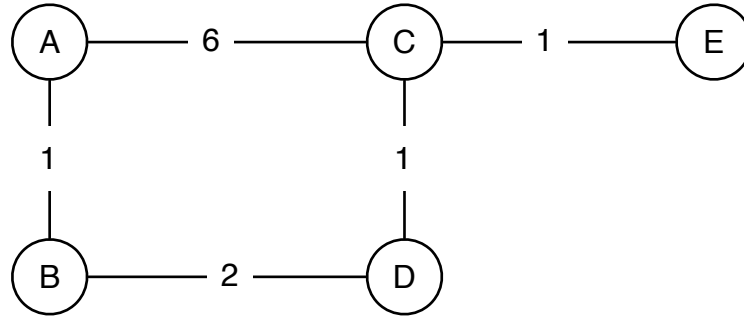
### 2.1 Johdatus reititykseen

Tietoliikenneverkkojen perimmäinen tehtävä on välittää tietoa kahden tai useamman verkossa olevan solmun välillä. Viestien välityksessä keskeinen osa on reitityksellä, joka huolehtii viestien kulkemisesta verkossa ennalta määrättyjen sääntöjen mukaan tiettyä reittiä pitkin. Vaikka reitityksellä voidaan tarkoittaa myös pakettien välittämistä solmujen välillä, tässä työssä keskitytään reitityksen toiseen puoleen: reittien selvittämiseen verkossa.

Reititys on monimutkainen ongelma, eikä sen suorittamiseen ole olemassa yhtä oikeaa ratkaisua. Tästä johtuen reitityksestä on olemassa useita muunnelmia eri tilanteisiin, joista suurin ero on staattisen ja dynaamisen reitityksen välillä. Staattinen reititys tarkoittaa menetelmää, jossa verkon ylläpitäjä asettaa reitit laitteisiin käsin konfiguroinnin yhteydessä. Staattisen reitityksen yksinkertaisuuden ja edellisessä kappaleessa tehdyn rajauksen vuoksi työ keskittyy dynaamisen reitityksen toimintaan.

Dynaaminen reititys on itsenäinen prosessi, joka selvittää reitit solmujen välillä nykyisessä verkossa ja mukautuu verkossa tapahtuviin muutoksiin. Dynaamiset reititysprotokollat voidaan karkeasti jakaa kahteen ryhmään: etäisyysvektori-protokolliin ja linkkitilaprotokolliin. Tässä luvussa käydään läpi molempien ryhmien toimintaperiaatteet yleisellä tasolla.

Kuvassa 1 esitetään yksinkertainen tietoliikenneverkko. Kuvan ympyrät esittävät reitittimiä, ja ympyröiden välillä olevat viivat kuvaavat reitittimien välisiä linkkejä verkossa. Viivojen päällä olevat numerot ovat kyseisen linkin kustannuksia eli hintoja: pienempi numero tarkoittaa hinnalta halvempaa eli reitityksellisesti suositellumpaa linkkiä. Tässä luvussa käytetään kuvan 1 mukaista verkkoa havainnollistamaan graafiteorian käsitteitä ja protokollien toimintaa.



Kuva 1: Yksinkertainen esimerkkiverkko

## 2.2 Graafiteoria

Teoreettisella tasolla tietoliikenneverkkoja mallinnetaan graafeina. Graafi  $G$  koostuu kahdesta joukosta: solmuista  $V$  ja solmuja yhteenliittävistä kaarista  $E$  siten, että  $E$  koostuu 2-alkioisista joukoista  $V$ :n alkioita. Matemaattisesti ilmaistuna

$$G = (V, E), E \subseteq [V]^2 \quad (1)$$

Yleinen tapa esittää graafi on piirtää piste jokaista solmua kohden ja yhdistää joukosta  $E$  löytyvät solmuparit keskenään viivalla. Solmujoukon  $V$  ja kaarien  $E$

$$V = \{A, B, C, D, E\}, E = \{\{A, B\}, \{A, C\}, \{B, D\}, \{C, D\}, \{C, E\}\} \quad (2)$$

muodostama graafi voidaan esittää kuvan 1 mukaisena verkkona.

Graafin  $G$  kaksi solmua  $x_0$  ja  $x_1$  ovat naapureita, jos  $\{x_0, x_1\}$  on graafista  $G$  löytyvä kaari. Polku on graafi  $P = (V, E)$ , jossa

$$V = \{x_0, x_1, x_2, \dots, x_k\}, E = \{\{x_0, x_1\}, \{x_1, x_2\}, \dots, \{x_{k-1}, x_k\}\} \quad (3)$$

solmut  $x_0$  ja  $x_k$  yhdistyvät polkua  $P$  pitkin. Kuvan 1 esittämä graafi  $G$  sisältää esimerkiksi polun

$$P = (V, E), V = \{A, C, D\}, E = \{\{A, C\}, \{C, D\}\} \quad (4)$$

Kaaren hinta on käänteinen suosimisarvo kaaren yli kulkemiselle.

Lyhimmän polun ongelma tarkoittaa polun löytämistä kahden solmun välillä niin, että polun kaarien hintojen summa on pienin mahdollinen. Matemaattisesti ilmaistuna polun  $P = (\{x_0, x_1, \dots, x_k\}, \{\{x_0, x_1\}, \dots, \{x_{k-1}, x_k\}\})$  hinta  $W(P)$  on

polun varrella olevien kaarien hintojen summa:

$$W(P) = \sum_{i=1}^k W(x_{i-1}, x_i) \quad (5)$$

Lyhimmän polun  $x_n \rightsquigarrow x_m$  hinta  $\delta(x_n, x_m)$  on

$$\delta(x_n, x_m) = \begin{cases} \min\{W(P) : x_n \rightsquigarrow x_m\}, & \text{jos polku } (x_n, x_m) \text{ on olemassa} \\ \infty, & \text{muuten} \end{cases} \quad (6)$$

Lyhyin polku solmulta  $x_n$  solmulle  $x_m$  on mikä tahansa polku  $P$ , jonka hinta  $W(P) = \delta(x_n, x_m)$ . [1]

Reitityksen kannalta lyhimmän polun ongelma on graafiteorian kiinnostavin osa, ja tietoliikenteen reititys perustuu nimenomaan tämän ongelman ratkaisemiseen. Kaksi yleisesti käytössä olevaa algoritmia tämän ongelman ratkaisuun ovat Bellman-Ford ja Dijkstra.

## Bellman-Ford

Bellman-Ford on Richard Bellmanin ja Lester Ford Jr. vuosina 1958 ja 1956 julkaisema algoritmi [2] [3]. Se ei ole niin tehokas kuin muut saman ongelman ratkaisevat algoritmit, mutta Bellman-Fordin etuna graafiteoriassa on sen kyky toimia oikein kaarien negatiivisten hintojen kanssa. Bellman-Fordin toiminta perustuu suurelta osin solmujen relaksointiin (engl. relaxing). Relaksoinnissa selvitetään, onko solmua mahdollista saavuttaa lyhyempää reittien kuin tällä hetkellä on tiedossa. Alle oleva pseudokoodi esittää sekä algoritmin toiminnan että relaksoinnin. `relax()`-funktio ottaa parametrina kaksi solmua  $u$  ja  $v$ , sekä näiden välisen hinnan. Jos algoritmin tällä hetkellä tiedossa oleva hinta  $v$ -solmulle on suurempi kuin hinta  $u$ :lle lisättyinä  $u:n$  ja  $v:n$  väliseen hintaan, päivitetään uusi hinta ja asetetaan tieto, että  $v$  on saavutettavissa  $u:n$  kautta. Relaksointi suoritetaan graafin jokaiselle kaarelle  $|V| - 1$  kertaa, missä  $|V|$  on solmujen lukumäärä graafissa. Näiden suorituskertojen aikana lasketut lyhimät polut lähenyvät ja lopulta saavuttavat oikean vastauksen. Lopuksi algoritmi tarkistaa onko graafissa mukana lähtösolmulta saavutettavissa oleva negatiivinen sykli. Mikäli ei ole, se palauttaa onnistunutta laskemista kuvaavan tosiarvon, mutta mikäli negatiivinen sykli löytyy, palautetaan `false` joka kertoo, ettei lyhimpiä polkuja voitu laskea. Algoritmin suoritusaika  $O$ -notaatioissa on  $O(VE)$ . [1]

```

relax(u,v,w):
    if v.d > u.d + w:
        v.d = u.d + w;
        v.p = u;

bellmanford(G,w,s):
    initialize-single-source(G,s);

    for i = 1 to |G.V| - 1:
        for each edge(u,v) in G.E:
            relax(u,v,w(u,v));

    if contains negative cycle return false
    else return true

```

Initialize-single-source() funktio asettaa kaikkien solmujen hinnaksi äärettömän lukuunottamatta lähdesolmua, jonka hinnaksi asetetaan nolla.

Bellman-Fordin hajautettua versiota käytetään yleisesti reitinlaskentaan etäisyysvektori-protokollien kanssa, koska toisin kuin esimerkiksi Dijkstran algoritmi, Bellman-Fordin ei tarvitse tietää täydellistä verkon topologiaa reittien laskemiseksi. Luvussa 2.3 tutustutaan tarkemmin etäisyysvektori-protokollien toimintaan.

## Dijkstran algoritmi

Dijkstran algoritmi on alankomaalaisen Edsger Dijkstran vuonna 1959 julkaisema lyhimmän polun ongelman ratkaiseva algoritmi [4]. Algoritmille annetaan parametrisi kuva verkosta (graafi) sekä haluttu laskennan aloitussolmu. Dijkstra laskee syötteestään lyhimmän (halvimman) mahdollisen reitin aloitussolmulta kaikkiin muihin verkon solmuihin. Dijkstran toimintaperiaate on yksinkertainen: otetaan halvin tunnettu kohde jota ei ole vielä käsitelty, ja relaksoidaan kaikki kyseisen kohteen naapurit kuten Bellman-Fordissa. Alla olevassa pseudokoodissa esitetään algoritmin toimintaperiaate. Mikäli kyseisessä pseudokoodissa oleva  $Q$  on toteuttu fibonacci-kekona, saadaan algoritmin suoritusajaksi  $O$ -notaatioissa  $O(E + V \log V)$ . [1]

```

dijkstra(G,w,s):
    initialize-single-source(G,s);
    S = 0;
    Q = G.V;
    while Q != 0:
        u = extract-min(Q)
        S = S + u
        for each vertex v in G.Adj[u]:
            relax(u,v,w(u,v))

```

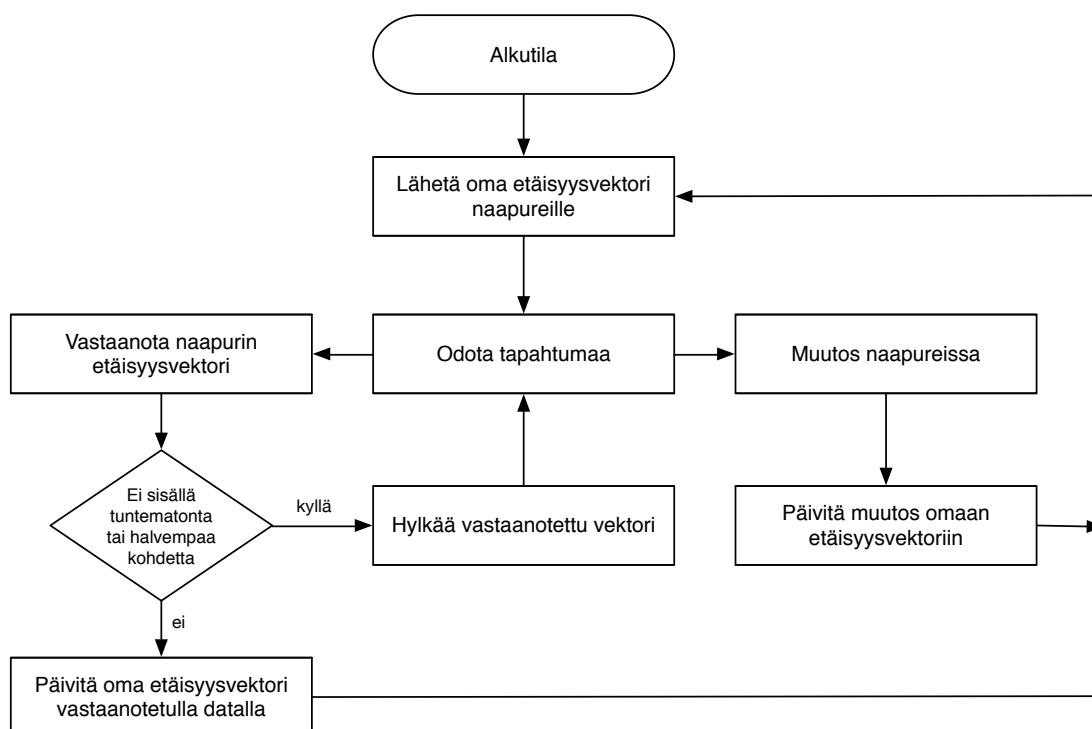
Kuten etäisyysvektoreidenkin tapauksessa, `initialize-single-source()` funktio asettaa kaikkien solmujen hinnaksi äärettömän paitsi lähdesolmun, jonka hinnaksi asetetaan 0.

Dijkstran algoritmia käytetään yleisesti reitinlaskentaan linkkitilaprotokollien kanssa, koska se tarjoaa tehokkaan ratkaisun lyhimmän polun ongelmaan kun verkon topologia on tiedossa.

## 2.3 Etäisyysvektori-protokollat

Etäisyysvektori-protokollissa jokainen reititin pitää yllä tietokantaa verkon muista reitittimistä: minkä naapurin kautta kohde saavutetaan ja mikä on hinta kyseiselle reitittimelle. Tätä tietokantaa kutsutaan reitittimen etäisyysvektoriksi. Kun reititin lähettää etäisyysvektorinsa eteenpäin, vastaanottava reititin kasvattaa siinä olevia etäisyyksiä vastaanotetun linkin hinnan verran, ja merkitsee miltä naapurilta kyseinen vektori vastaanotettiin. Jos vastaanotettu vektori ilmoittaa etäisyyden tiettyyn solmuun halvemmaksi kuin reitittimen oma etäisyysvektori, päivittää reititin oman vektorinsa ja lähettää sen eteenpäin naapureilleen, jotka jälleen käsittelevät vastaanottamansa vektorin ja päivittävät omansa tarpeen mukaan.

Yksinkertaistettu tilakone etäisyysvektori-protokollien toiminnasta on esitetty kuvassa 2. Protokollan käynnistyessä alkutilasta lähetetään ensin oma etäisyysvektori naapureille, minkä jälkeen siirrytään odottamaan ulkoista tapahtumaa. Ulkoisia tapahtumia ovat naapurin lähettämän etäisyysvektorin vastaanotto ja muutos omista naapureista. Kun naapurin etäisyysvektori vastaanotetaan, tutkitaan sisältääkö kyseinen vektori tuntematonta tai halvempaa kohdetta kuin oma vektori. Jos ei, hylätään vastaanotettu vektori ja jäädään odottamaan uutta tapahtumaa. Jos uusi tai halvempi kohde löytyy vastaanotetusta vektorista, päivitetään oma vektori vastaamaan kyseistä vektoria ja lähetetään oma vektori eteenpäin verkossa omille naapu-



Kuva 2: Etäisyysvektoriprotokollien yksinkertaistettu tilakone

reille. Tämän jälkeen jäädään jälleen odottamaan uutta tapahtumaa. Muutos naapuritilassa on etäisyysvektorin vastaanottamista yksinkertaisempi tapahtuma. Kun tieto naapurin tilan muutoksesta vastaanotetaan, päivitetään oma vektorimme ja välitetään se eteenpäin naapureillemme, minkä jälkeen jäädään jälleen odottamaan uutta tapahtumaa.

Reitityksen käynnistyessä kuvan 1 mukaisessa verkossa ajanhetkellä 0 jokainen reititin tietää verkosta ainoastaan itsensä. Tilanne on kuvattu taulukossa 1. Jokainen reititin lähettää oman etäisyysvektorin naapureilleen, jolloin kaikkien tiedossa on ensimmäisen vaiheen jälkeen omat suorat naapurit. Vektorit tulvitetaan eteenpäin, jolloin verkko konvergoituu välivaiheiden kautta lopputilanteeseen. Huomionarvoista on, miten solmun A reititystaulussa etäisyys kohteeseen E tippuu ajanhetken 1 hinnasta 6 ajanhetken 3 hintaan 5 etäisyysvektoreiden levittyä koko verkkoon.

Etäisyysvektoreihin perustuvat protokollat ovat yksinkertaisia toteuttaa, mutta niiden toiminta on suuremmissa verkoissa tehotonta. Ne myös kärsivät tunnetuista haasteista tietyissä tilanteissa, esimerkiksi verkon jakautuminen on ongelmallista. Siihen on kuitenkin kehitetty ratkaisuja, joita ovat jaettu horisontti ja jaettu horisontti myrkytetyllä vektorilla. Jaetun horisontin tilanteessa solmu ei mainosta reittejä takaisin reitittimelle josta se oppi kyseisen reitin. Esimerkiksi kuvan kuvan 1



mukaisessa verkossa jaetun horisontin tapauksessa solmu C ei kerro etäisyysvektorisissaan solmun D suuntaan tietävänsä mitään solmusta B, koska jos linkit BD ja AC katkeavat, mainostaisi C virheellisesti B:lle että solmu B (ja A) on saavutettavissa tätä kautta. Myrkytetty vektori lisää vielä tehokkuutta verkon jakautuessa siten, että C mainostaa D:lle etäisyyttä kohteelle B äärettömäksi, jolloin D tietää välittömästi ettei B:tä voida saavuttaa solmun C kautta.

Taulukko 1: Solmun C reititystaulut eri ajanhetkillä

Ajanhetki 0			Ajanhetki 1			Ajanhetki 2			Ajanhetki 3		
N	SH	H	N	SH	H	N	SH	H	N	SH	H
C	-	0	C	-	0	C	-	0	C	-	0
			A	A	6	A	A	6	A	D	4
			D	D	1	D	D	1	D	D	1
			E	E	1	E	E	1	E	E	1
						B	D	3	B	D	3

Taulukossa 1 esitetään kuvan 1 mukaisen verkon solmun C reititystaulut eri ajanhetkillä. Taulukosta huomataan, että ajanhetkillä 1 ja 2 halvin tunnettu reitti A:lle on hinnaltaan 6, mutta ajanhetkellä 3 reitti kohteelle A päivittyy nopeammaksi, kun D:n etäisyysvektori jossa A on mukana B:n kautta saavutettuna saapuu C:lle.

Tunnetuin etäisyysvektoriprotokolla on Routing Information Protocol (RIP) [5]. Border Gateway Protocol (BGP) [6] on polkuvektoriprotokolla, joka ilmoittaa etäisyysvektorin sijaan koko polun kohdeverkolle. Tällä toiminnalla pyritään estämään reitityssilmukoiden syntyminen.

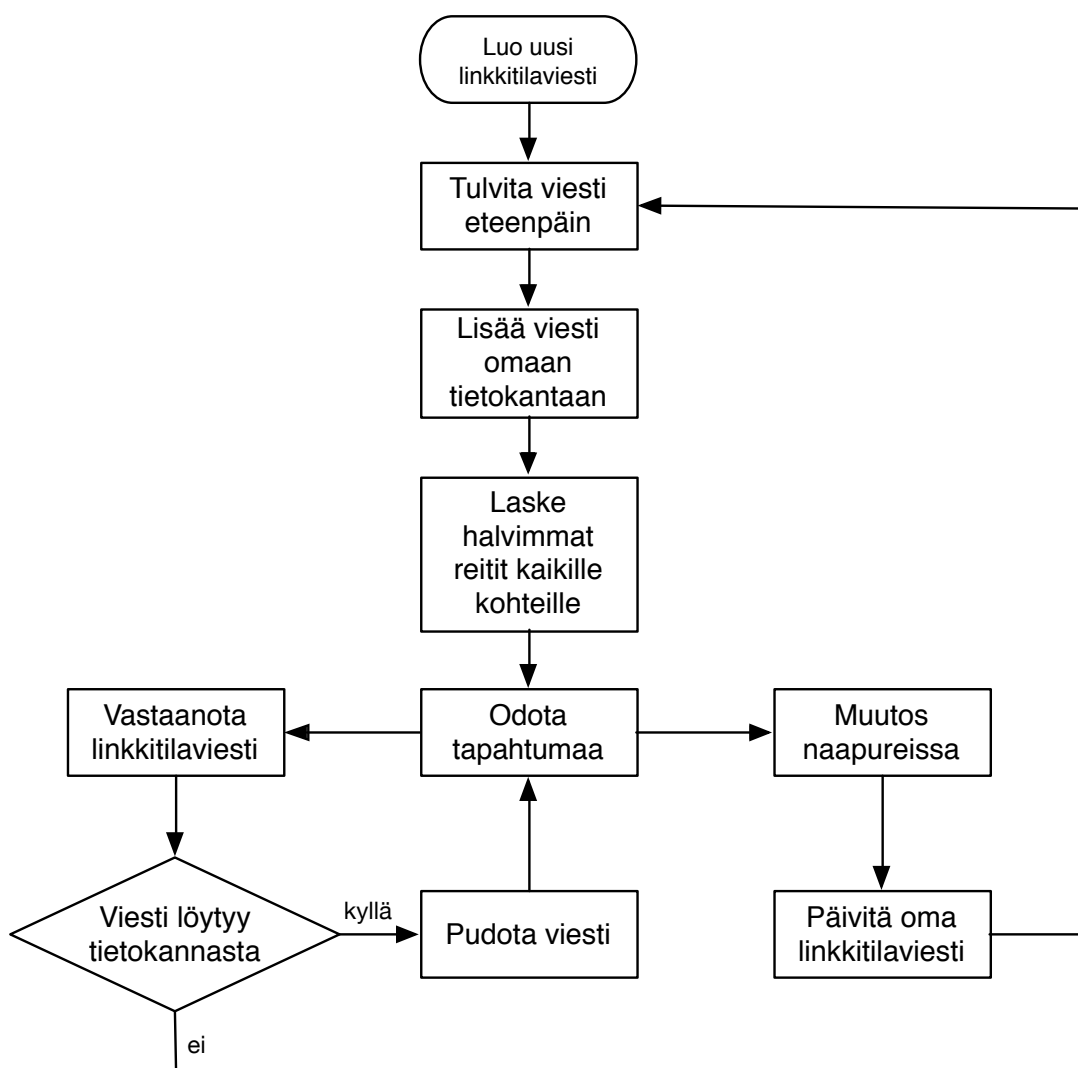
## 2.4 Linkkitilaprotokollat

Linkkitilaprotokollat perustuvat jokaisen reitittimen naapuritiedon (eli linkkitilan) levittämiseen koko verkkoon. Kun jokainen verkon reititin tietää jokaisen reitittimen naapurit, voidaan tiedon perusteella muodostaa täydellinen verkkokuva ja laskea lyhyimmät mahdolliset reitit eri reitittimien välillä verkossa. Taulukossa 2 on esitettyä esimerkit reitittimien generoimista linkkitilaviesteistä kuvan 1 verkossa. Esimerkiksi kyseissä verkossa solmulla A on naapureinaan B ja C, joiden hinnat A:n kautta saavutettaessa ovat 1 ja 6.

Jokaisen reitittimen linkkitilaviesti levitetään verkkoon tulvittamalla, ja jokainen reititin kokoaa oman linkkitilatiekoksensa tallentamalla kaikkien verkon reitittimien linkkitilaviestit. Linkkitilatiekokannan perusteella reitittimet voivat laskea lyhyimmät (halvimmat) reitit kaikkiin verkon reitittimiin. Laskennan tulokset tallennetaan reititystauluun, jonka perusteella reitittimelle saapuvia paketteja ohjataan

Taulukko 2: Esimerkki reitittimien yksinkertaistetuista linkkitilaviesteistä

Solmu A	Solmu B	Solmu C	Solmu D	Solmu E																														
<table><tr><th>N</th><th>H</th></tr><tr><td>B</td><td>1</td></tr><tr><td>C</td><td>6</td></tr></table>	N	H	B	1	C	6	<table><tr><th>N</th><th>H</th></tr><tr><td>A</td><td>1</td></tr><tr><td>D</td><td>2</td></tr></table>	N	H	A	1	D	2	<table><tr><th>N</th><th>H</th></tr><tr><td>A</td><td>6</td></tr><tr><td>D</td><td>1</td></tr><tr><td>E</td><td>1</td></tr></table>	N	H	A	6	D	1	E	1	<table><tr><th>N</th><th>H</th></tr><tr><td>B</td><td>2</td></tr><tr><td>C</td><td>1</td></tr></table>	N	H	B	2	C	1	<table><tr><th>N</th><th>H</th></tr><tr><td>C</td><td>1</td></tr></table>	N	H	C	1
N	H																																	
B	1																																	
C	6																																	
N	H																																	
A	1																																	
D	2																																	
N	H																																	
A	6																																	
D	1																																	
E	1																																	
N	H																																	
B	2																																	
C	1																																	
N	H																																	
C	1																																	



Kuva 3: Linkkitilaprotokollien yksinkertaistettu tilakone

eteenpäin halvinta mahdollista reittiä pitkin. Yksinkertaistettu tilakone linkkitilaprotokollien toiminnasta on esitetty kuvassa 3. Tilakoneen toiminta lähtee liikkeelle luomalla oma linkkitilaviesti, joka tulvitetaan naapureille. Tämän jälkeen viesti lisätään omaan linkkitilatiekantaan, jonka perusteella lasketaan halvimmat mahdol-

liset reitit kaikille tietokannasta löytyville solmuille. Reitinlaskennan jälkeen odotetaan ulkoisia tapahtumia, joita ovat linkkitilaviestin vastaanotto ja muutos omissa naapureissa, eli omassa linkkitilassa. Linkkitilaviestiä vastaanotettaessa tutkitaan, löytyykö kyseinen viesti jo omasta linkkitilatietokannasta. Jos viesti löytyy, se tiputetaan ja jäädään odottamaan uutta tapahtumaa. Jos viestiä ei löydy, se tulvitetaan eteenpäin verkossa eli lähetetään kaikille muille, paitsi viestin solmulle lähettäneelle naapurille. Tulvituksen jälkeen viesti lisätään tietokantaan ja suoritetaan uusi reitinlaskenta, minkä jälkeen jäädään odottamaan uutta tapahtumaa. Jos tapahtumana on muutos naapureissa, päivitetään oma linkkitilaviesti joka lähetetään eteenpäin kaikille naapureille, lisätään linkkitilatietokantaan ja suoritetaan reitinlaskenta uudelleen.

Linkkitilatiedon levitykseen perustuvat protokollat ovat monimutkaisempia toteuttaa verrattuna etäisyysvektori-protokolliin, mutta ne skaalautuvat ja konvergoituvat paremmin isoissa verkoissa. Tunnetuimpia linkkitilanprotokollia ovat Intermediate System to Intermediate System (IS-IS) [7] ja Open Shortest Path First (OSPF) [8]. IS-IS on alunperin OSI-tietoverkoille kehitetty protokolla, joka on joustavasta rakenteestaan johtuen yksinkertainen käyttää myöskin muissa ympäristöissä, esimerkiksi IP-verkoissa. OSPF on tarkoitettu ainoastaan IP-verkkojen reititykseen, ja protokolla onkin matalalta tasolta asti liitetty tiukasti yhteen IP-protokollan kanssa.

## 2.5 Reititys internetissä

Internet on maailmanlaajuinen tietoliikenneverkko, jolla on jo lähes kolme miljardia käyttäjää [9]. Tämän kokoisessa verkossa on mahdotonta käyttää yhtä reitityspotentiaalia, joka tietäisi aina mistä päin verkkoa kaikki mahdolliset kohteet löytyvät.

Internet Protocol (IP) [10] on tärkein protokolla internetin toiminnan kannalta. Internet yhdistää lukemattomia erilaisia verkkotekniikoita ja sovelluksia keskenään, ja IP on ainoa yhteinen tekijä koko internetissä. IP-osoite koostuu 32-bitistä, ja se usein esitetään tavu kerrallaan pisteillä erotettuna desimaalimuodossa, esimerkiksi 10.17.100.2 on oikein muodostettu IP-osoite. IP-osoite koostuu kahdesta eri osasta: verkko-osoitteesta ja kohdeosoitteesta. Alunperin IP-osoitteessa oli kiinteä 8 bitin verkko-osa, mutta verkon kasvaessa siirryttiin käyttämään hieman joustavampaa luokkajaottelua: luokan A-osoitteissa oli 8 bitin, B-osoitteissa 16 bitin ja C-osoitteissa 24 bitin verkko-osa. Kasvun jatkuessa ei tämäkään jaottelu riittänyt, vaan käyttöön otettiin luokaton reititys (engl. Classless Inter-Domain Routing, CIDR) [11]. Luokaton reititys poistaa luokkajaottelun IP-osoitteista ja ottaa käyttöön verkkopeitteen käsitteen. Verkkopeitteellä ilmoitetaan, kuinka mon-

ta bittiä osoitteesta viittaa verkko-osoitteeseen, jolloin loput bitit jäävät kohdeosoitteiden käyttöön. Verkkopeite voidaan ilmoittaa kahdella eri tavalla: käyttämällä samanlaista notaatiota kuin varsinaisten IP-osoitteiden kanssa, eli esimerkiksi 255.255.252.0. Kun verkkopeite muutetaan bittimuotoon, saadaan 1111 1111.1111 1111.1111 1100.0000 0000. Tästä muodosta nähdään, että kyseistä verkkopeitettä käyttävän osoitteen 22 ensimmäistä bittiä kuvaa verkko-osoitetta. Sama asia voidaan ilmoittaa myös toisella tavalla käyttämällä /-notaatiota. Esimerkiksi 10.17.100.2/22 tarkoittaa edellisen esimerkin kanssa täysin samaa asiaa eri tavalla ilmaistuna.

Verkon suuresta koosta, dynaamisuudesta ja kasvuvauhdista johtuen internet on jaettu useisiin pieniin, keskitetysti ylläpidettyihin verkkoalueisiin. Näitä verkkoalueita kutsutaan nimellä autonominen järjestelmä (engl. Autonomous System, AS). Reititys autonomisen järjestelmän sisällä tapahtuu käyttämällä esimerkiksi OSPF [8], IS-IS [7] tai RIP -protokollaa. Autonomisen järjestelmän ulkopuolelle mainostetaan luokatonta reititystä käyttämällä ainoastaan sisältä löytyviä verkko-osoitteita. Näin säästetään huomattavasti tilaa reititystauluissa ympäri internetiä.

## 2.6 Erityisvälitysverkot

Erityisvälitysverkot ovat tietoverkkoja, joissa viiveet ja pakettihäviöt voivat olla huomattavan suuria, eikä niillä välttämättä ole ennaltamääriteltyä hierarkiaa. Tässä luvussa määritellään mitä ovat erityisvälitysverkot, kuvaillaan niiden ominaispiirteitä ja käydään läpi erityisesti haasteelliseen olosuhteisiin suunniteltuja lähestymistapoja reititykseen.

### Delay Tolerant Network

Viivesietoiset verkot (engl. Delay Tolerant Networking, DTN) ovat alunperin planeettojen välisen liikennöinnin (engl. InterPlanetary Networking, IPN) [12] tutkimuksesta kehittyneitä verkkoratkaisuja, joissa perinteiset internetissä käytössä olevat protokollat eivät toimi suuresta viiveestä ja pakettihäviöstä johtuen. Kevin Fall esitteli vuonna 2003 julkaistussa artikkelissa [13] yleiskäyttöisemmän viivesietoisen verkon arkkitehtuurimallin haasteellisille verkoille. Julkaisussa luetellaan ongelmallisia oletuksia, joita TCP/IP-pohjaiset ratkaisut asettavat verkolle: jatkuva päästä-päähän yhteys, kohtuullisen matalat viiveet ja pieni pakettihäviö. Haasteelliset verkot voivat kuitenkin rikkoa näitä oletuksia, mistä johtuen TCP/IP-pohjaiset ratkaisut eivät toimi näissä ympäristöissä. Fall listaa esimerkeiksi haasteellisista verkoista maanpäälliset liikkuvat verkot (lyhyen kantaman radion sisältämä välittäjänä toimiva linja-auto), erikoiset välitysverkot (hyvin suuret viiveet säännöllisin katkoin,

esimerkiksi avaruusviestintä), sotilasverkot (voimakas häirintä, haasteelliset maasto-olosuhteet, korkeamman prioriteetin liikenne) ja sensoriverkot (solmujen kapasiteetti erittäin rajoitettu). Näiden pohjalta määritellään haasteellisten verkkojen käsite, jota käytetään pohjana paperissa esitetylle arkkitehtuurimallille. Määritelmä on jaettu kolmeen osaan: linkkien ja polkujen ominaisuuksiin (korkeat viiveet, katkonaisuus ja pitkät jonotusajat), verkkojen arkkitehtuuriin (yksinkertaiset ja rajoittuneet siirrotekniikat eivät toimi muiden tekniikoiden kanssa yhteen, ja verkon pienen kapasiteetin käyttöä tulee rajoittaa käyttäjähallinnan kautta) ja solmujen ominaisuuksiin (solmujen mahdollisesti lyhyen elinikä ja rajoitetut resurssit). [13]

Suurelta osin Fallin työn pohjalta on kehitetty RFC 4838 Delay-Tolerant Networking Architecture [14]. Tässä dokumentissa esitellään edelleenkehitetty malli viivesietoisten verkkojen arkkitehtuuriksi. Esitetyssä mallissa luodaan eri verkkotekniikoiden päälle yhtenäinen sanomapohjainen päällysverkko, jossa sanomien välitys ja reitittäminen tapahtuu alla toimivista yksittäisistä verkkotekniikoista välittämättä. Tätä päällysverkkoa kutsutaan pakkakerrokseksi (Bundle layer), ja päällysverkon toteuttavia solmuja kutsutaan viivesietoisiksi solmuiksi (DTN nodes). [14]

DTN-arkkitehtuuri ei ota suoraan kantaa miten reititys tulisi toteuttaa, vaan tarjoaa arkkitehtuurin jonka päälle voidaan toteuttaa reititys halutunlaiseksi. Reititys viivesietoisissa verkoissa voidaan toteuttaa siis useilla eri tavoilla. Korkealla tasolla reititystavat voidaan jakaa kahteen osaan: deterministiseen ja stokastiseen reititykseen. Deterministinen reititys sisältää esimerkiksi SPF-algoritmiin ja puupohjaisiin rakenteisiin perustuvat ratkaisut. Stokastinen reititys taas perustuu sanomien satunnaiseen tai valistuneeseen arvaukseen perustuvaan tulvittamiseen. Oraakkeliksi kutsutaan stokastisessa reitityksessä solmua, jolla on tiedossaan verkon nykyinen ja tuleva tila, jota voidaan hyödyntää reitityksessä. Oraakkeleita voi olla useita eri tyyppisille tiedoille, esimerkiksi solmujen väliset yhteydet voivat olla yhden oraakkelin tiedossa ja solmujen puskurikoot toisella. Oraakkeliin toteuttaminen käytännössä on kuitenkin vaikeaa. Käytännöllisempää onkin laskea verkon historiasta todennäköisyyksiä esimerkiksi tuleville yhteyksille.

Jos verkolla ei ole tarkkaa tietoa tilastaan, täytyy sanomaa tulvittaa eteenpäin ja olettaa, että sanoma joskus päätyy kohteelleen. Tätä mallia kutsutaan epidemia-reititykseksi [15]. Sanoman tärkeydestä ja käytössä olevasta verkon tilan tiedosta riippuen voidaan viestin tulvittamista verkossa optimoida valitsemalla tarkemmin solmut mihin tulvitus suoritetaan.

Koska DTN-verkoissa ei oleteta, että kahden solmun välillä olisi koskaan suoraan toimivaa päästä-päähän yhteyttä, toimivat verkon solmut tarvittaessa sanomien välittäjinä. Tätä toiminnallisuutta kutsutaan talleta-kanna-välitä (engl. Store-Carry-

Forward, SCF) -menetelmäksi. Vastaanottaessaan uuden sanoman solmu tallentaa sen muistiin, ja kohdatessaan uuden solmun kantaja välittää sanoman eteenpäin. Sanomien välitykseen vaikuttaa luonnollisesti käytössä oleva reititysprotokolla.

Haasteellisia verkkoja voidaan mallintaa dynaamisina verkkoina, jotka jaetaan verkon luoteen perusteella kolmeen pääluokkaan. Lopulta yhdistyvä dynaaminen verkko (engl. Eventually Connected Dynamic Network, ECDN) on verkko, joka jossa kaikkien verkon solmujen välillä on olemassa polku samalla ajanhetkellä  $t$ . Lopulta reititettävässä dynaamisessa verkossa (engl. Eventually Routable Dynamic Network, ERDN) löytyy reitti ajanhetkellä  $t$  lähettävän ja vastaanottavan reitin välillä. Lopulta välitettävä dynaaminen verkko (engl. Eventually Transportable Dynamic Network, ETDN) hoitaa sanomien välityksen välisolmujen kautta. Vaikka yhteinäistä reittiä lähettävän ja vastaanottavan solmun välillä ei olisi missään vaiheessa, pystyy verkko silti välittämään sanoman perille. [16]

## Mobile Ad Hoc Network

Mobile Ad Hoc Network (MANET) on itsenäisistä liikkuvista jäsenistä koostuva langaton verkko, jolla ei ole ennaltamääritettyä hierarkiaa tai keskitettyä ylläpitoa. MANET-tutkimus oli aktiivista 90-luvulla samaan aikaan DTN-verkkojen edeltäjän, planeettojen välisten verkkojen (IPN) tutkimuksen kanssa, mutta tutkimukset tapahtuivat kuitenkin täysin erillisinä. [17] [18]

Verkon jäsenet kommunikoivat suoraan radionsa kantoalueella olevien jäsenten kanssa, ja oman radion kantoalueen ulkopuolelle liikennöidessä verkon jäsenet toimivat viestien välittäjinä. Tämän tyyppinen verkko vaatii keskenään liikennöiviltä jäseniltä toimivan päästä-päähän yhteyden. Tästä johtuen verkot on mahdollista toteuttaa IP-pohjaisina. Dynaamisista verkoista MANET-verkot vastaavat ERDN-verkkoja. MANET-verkkojen reititys voidaan jakaa kolmeen pääkategoriaan: ennakoiviin, vastavaikutteisiin ja risteäviin reititysprotokolliin.

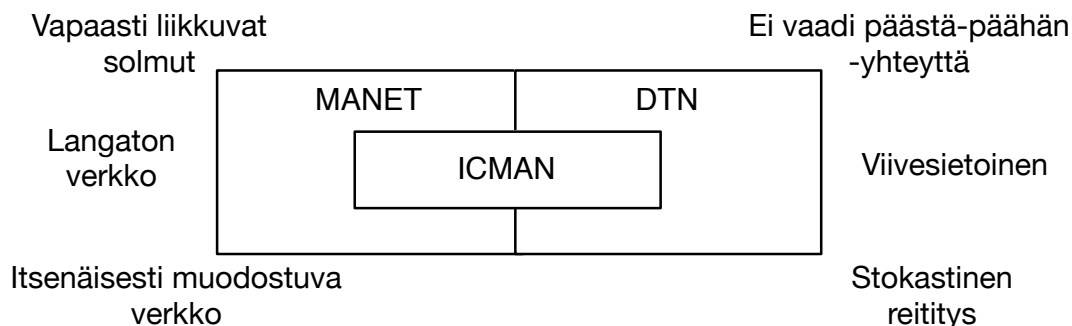
Ennakoivassa reitityksessä (engl. proactive routing) verkoissa verkon jäsenet lähettävät toistuvasti jaksottain reititystietoa, jonka perusteella jokainen verkon jäsen voi ylläpitää ajantasaista reititystaulua. Tyypillinen esimerkki ennakoivasta reititysprotokollasta MANET-verkoissa on Optimized Link State Routing (OLSR) [20].

Vastavaikutteiset reititysprotokollat (engl. reactive on-demand routing) etsivät reitin lähteeltä kohteelle vasta tarvittaessa. Kun reitti on kerran luotu, muistaa verkko reitin kunnes se ei ole enää ajantasainen. Ad Hoc On-Demand Distance Vector (AODV) reititys on yksi tähän kategoriaan kuuluvista protokollista [20].

Risteävät reititysprotokollat (engl. hybrid routing protocol) hyödyntävät ominaisuuksia sekä ennakoivista että vastavaikutteisista protokollista. Yksi esimerkki ris-

teävästä protokollasta on Zone Routing Protocol (ZRP), joka muodostaa läheisiin jäseniinsä reitit ennakoivien protokollien tapaan, ja kaukaisiin taas vastavaikutteisesti.

### Intermittently Connected Mobile Ad Hoc Network



Kuva 4: ICMAN yhdistää MANET- ja DTN-verkkojen ominaisuuksia

MANET-verkot olettavat toistensa kanssa kommunikoivilta jäseniltään päästä-päähän -yhteydellisyyttä. Verkkoja, joissa ei ole tätä rajoitetta, kutsutaan kirjallisuudessa nimellä Intermittently Connected Mobile Ad Hoc Network (ICMAN). ICMAN yhdistää ominaisuuksia sekä MANET (vapaasti liikkuvat solmut, itsenäisesti muodostuva langaton verkko)- että DTN-verkoista (päästä-päähän -yhteyttä ei vaadita, stokastinen reititystapa, viivesietoinen) (kuva 4). Yhdistelmä toteutetaan yleisesti liittämällä MANET-verkkoon DTN-verkkojen tapa hyödyntää tietoa välittäviltä jäseniltä löytyvää tallennustilaa väliaikaisena sijoituspaikkana. Kun verkon topologia muuttuu jäsenten liikkuesssa, voidaan tietoa mahdollisesti välittää edelleen kohti vastaanottajaa SCF-menetelmää käyttämällä. Toisin kuin MANET-verkoissa, joissa solmujen liikkuminen on ongelmallista reitityksen kannalta, on ICMAN-verkoissa solmujen liikkuminen ensiarvoisen tärkeää jotta sanomat päätyvät vastaanottajalleen. SCF-menetelmän käyttö ICMAN-verkkojen kanssa voidaan jakaa kolmeen alakategoriaan: supersolmuihin ja klustereihin perustuviin lähestymistapoihin, sekä täysin ad hoc -pohjaisiin lähestymistapoihin. Dynaamisista verkoista ICMAN-verkot kuuluvat ETDN-verkkojen kategoriaan. [18]

Supersolmut ovat verkon tavallisiin jäseniin verrattuna tehokkaampia ominaisuuksiltaan, jolloin niiden kyky välittää viestejä tavallisten jäsenten välillä on parempi. Supersolmut voivat olla kiinteästi sijoitettuina tiettyihin pisteisiin, tai ne voivat liikkua erillään olevien jäsenryppäiden välillä kuljettaen viestejä. [18]

Klusteriratkaisut toimivat verkoissa, joissa jäsenten liikkuminen ei ole täysin satunnaista, vaan tietyt jäsenet kokoontuvat todennäköisesti tiettyyn paikkaan toistuvasti. Tämä malli toimii erityisesti jos verkon liikkuvat jäsenet ovat ihmisiä: samat henkilöt usein kerääntyvät päivittäin samoihin pisteisiin, esimerkiksi töihin tai kouluun. [18]

Täysin ad hoc -pohjaiset ratkaisut jaetaan vielä kahteen alakategoriaan: tulvittaviin ja niin sanottuihin yhden kopion ratkaisuihin. Tulvittavista ratkaisuista tunnetuin on epideminen reititys, jossa sanomaa välittävät jäsenet edelleenlähettävät viestiä kaikille vastaantuleville jäsenille. Tulvituksen määrää voidaan myös optimoida esimerkiksi jäsenten sijaintien odotusarvojen tai verkon topologian perusteella. [18]

Tulvittavat ratkaisut ovat verkon resurssien kannalta kalliita. Tulvitukselle vaihtoehtoinen toimintamalli on käyttää yhden kopion ratkaisua. Yksinkertaisin yhden kopion ratkaisu on yhden hypyn, yhden kopion malli, jossa jäsen lähettää viestin eteenpäin vain, jos viestin lopullinen vastaanottaja on sen suora naapuri. [18]

## 2.7 Erityisvälitysverkkojen reititysprotokollia

Tässä luvussa käydään läpi erityisvälitysverkoissa yleisesti käytettyjä ja tunnettuja reititysprotokollia yleisellä tasolla.

### Ad Hoc On-Demand Distance Vector Routing

Ad Hoc On-Demand Distance Vector Routing (AODV) [19] on vastavaikutteinen etäisyysvektoriprotokolla, joka on suunniteltu ja optimoitu toimimaan MANET-verkoissa. "Counting to infinity-ongelma on ratkaistu käyttämällä kohteille sekvenssinumerointia, jolla vältetään silmukat verkossa kaikissa tilanteissa.

AODV käyttää viestinnässään kolmea eri viestityyppiä: reittipyyntöä (engl. route request, RREQ), reittivastausta (engl. route replies, RREP) ja reittivirhettä (route error, RERR). Reittipyyntö lähetetään, kun yritetään viestiä kohteelle jolle ei löydy reittiä reititystaulusta. Viestiä tulvitetaan monilähetyksenä verkossa eteenpäin kunnes reitti kohteelle löytyy jonkin viestin vastaanottavan solmun reititystaulusta. Kyseinen solmu generoi reittivastauksen, joka lähetetään alkuperäisen reittipyyntöviestin lähettäjän suuntaan täsmälähetyksenä. Jokainen reittipyyntöviestin välittävä solmu tallettaa välimuistiinsa solmun jolta pyynnön sai, jotta voi tarvittaessa välittää reittivastauksen takaisin kohti pyynnön lähettäjää. Kun alkuperäinen pyynnön lähettäjä vastaanottaa reittivastauksen, lisää se reititystauluunsa merkinnän kohteeseen, ja lähettää viestinsä. Jos jokin yhteys verkossa katkeaa, generoidaan reitti-



virheviesti, jolla ilmoitetaan muille solmuille tapahtuneesta muutoksesta.

### **Optimized Link State Routing protocol**

OLSR [20] on langattomiin verkkoihin optimoitu ennakoiva linkkitilareititysprotokolla. OLSR-protokollalle on ominaista rajoittaa viestien tulvittamista verkossa valitsemalla Multipoint Relay (MPR) -solmuja, jotka hoitavat linkkitilan levitystä verkossa. OLSR selvittää verkon tarkan topologian verkosta kahden hypyn päähän itsestään, ja valitsee MPR-solmut siten, että jokainen solmun kahden hypyn takaisista naapureista on saavutettavissa MPR-solmujen kautta.

OLSR käyttää kahta eri viestityyppiä: Hello ja TC (Topology Control). Hello-viestejä käytetään naapuruuksien luomiseen, ja TC-viestejä topologiatiedon levittämiseen verkossa.

### **Zone Routing Protocol**

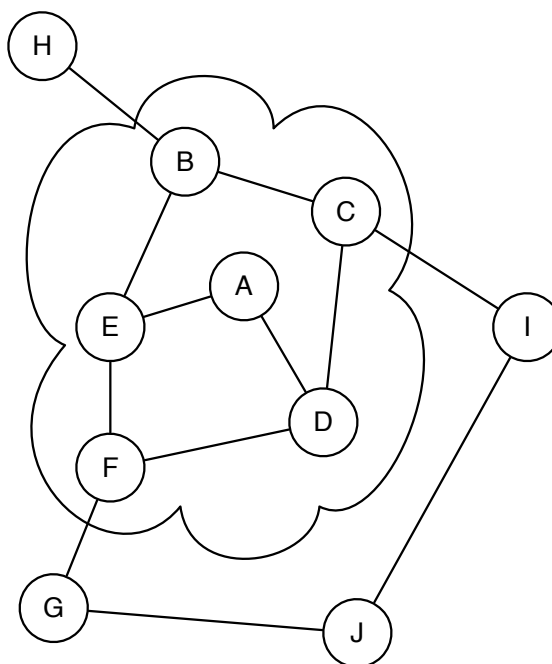
ZRP [21] on risteävä (engl. hybrid) reititysprotokolla, joka käyttää ennakoivaa reititystä läheisillä solmuille ja vastavaikutteista reititystä kauemmille solmuille. Ennakoivan reitityksen vyöhykkeen (engl. zone) koko on määritettävissä tilanteen mukaan riippuen verkosta: hyvissä ja vakaisissa verkoissa vyöhykettä kasvatetaan, ja vastaavasti huonossa verkossa sitä pienennetään. Kuvassa 5 on esitetty solmun A reititysvyöhyke, kun vyöhykkeen rajaksi on asetettu kaksi hyppyä.

Intrazone Routing Protocol (IARP) [22] vastaa ennakoivasta reitityksestä vyöhykkeen sisällä. Interzone Routing Protocol (IERP) [23] hoitaa vastaavasti reitityksen vyöhykkeen ulkopuolelle, kun reittiä kohdesolmulle ei ole tiedossa.

IARP ja IERP eivät ole mitään tiettyjä protokollia, vaan esimerkiksi ennakoiva linkkitilaprotokolla OLSR voi olla IARP ja vastavaikutteinen etäisyysvektori-protokolla AODV vastaavasti IERP.

### **Delay Tolerant Link State Routing**

Delay Tolerant Link State Routing (DTLSR) [24] on suunniteltu reititysprotokollaksi kehittyville alueille suuriviiveisiin verkkoihin. Sen toiminta-ajatus on hyvin lähellä perinteistä linkkitilaprotokollaa: jokainen verkon solmu tulvittaa verkkoon tiedon omista naapureistaan, ja jokainen solmu laskee itselleen reititystaulun vastaanottamiensa reityssanomien perusteella käyttäen Dijkstran algoritmia. Tulvitusta verkossa rajataan lainaamalla OSPF-protokollan aluemallia: jokainen solmu kuuluu tiettyyn alueeseen ja tulvittaa linkkitilasanomia ainoastaan oman alueensa sisällä. Muille alueille solmu mainostaa toimivansa yhdyskäytävänä tuntemilleen solmuille.



Kuva 5: Esimerkki ZRP-protokollan reititysvyöhykkeestä

Tavallisista linkkitilaprotokollasta poiketen DTLSR ei hoida itse naapureiden tilan tarkkailua, vaan luottaa alla olevan DTN-päällysverkon ilmoittamaan tietoon naapureiden tilasta. [24]

Ensimmäinen iso ero normaaleihin linkkitilaprotokolliin verrattuna on linkkitilaviestien hyvin pitkä elinikä. Jos tavallisissa protokollissa viestien elinikä on minuuttiluokkaa (esimerkiksi 20 minuuttia), käyttää DTLSR viestien elinikänä tunteja tai jopa päiviä. Tästä johtuen DTLSR lähettää uusia sanomia hyvin harvoin jos verkko pysyy muuttumattomana. Vuonna 2007 kirjoitetussa paperissa DTLSR-sanomien oletuselinäksi sanotaan yksi vuosi, joten käytännössä sanomia lähetetään verkkoon vain jos solmun linkkitilassa tapahtuu muutos. [24]

Toinen iso muutos liittyy reitinlaskentaan. Perinteisissä protokollissa solmu on mukana reitinlaskennassa ainostaan jos se on laskentahetkellä saavutettavissa. DTLSR ottaa laskennassa huomioon myös tällä hetkellä saavuttamattomissa olevat solmut olettamalla, että linkki poistuneelle solmulle on vain väliaikaisesti pois käytöstä. Mitä kauemmin solmun poistumisesta on, sitä suurempi hinta katkenneelle linkille tulee laskennassa. [24]

## Multi-Policy Link State Routing

Multi-Policy Link State Routing (MPLSR) [25] on yhdistelmäprotokolla tulvittavien ja sanomaa välittävien verkkojen välillä. Tulvittavat protokollat saavuttavat kohteen suuremmalla todennäköisyydellä, mutta ne kuormittavat usein pienikapasiteettisia verkkoja enemmän. MPLSR käyttää näiden yhdistelmää: se tulvittaa välitettäviä sanomia sen verran että todennäköisyys sanoman läpimenoon kasvaa, ja sen jälkeen välittää näitä useampaa kopiota sanomasta kohti kohdetta. Kuinka paljon sanomaan tulvitetaan voi riippua sanoman tärkeysasteesta: matalan tärkeysasteen viesti voidaan välittää kohti kohdetta käyttämällä ainoastaan yhden kopion välitystä, mutta tärkeä viesti voidaan vastaavasti tulvittaa hyvinkin suureen osaan verkkoa.

## 2.8 Yhteenveto

Tietoliikenneverkkojen tarkoitus on välittää viestit verkossa lähettäjältä vastaanottajalle. Reitityksen tehtävänä on selvittää miten viestit välitetään verkon läpi optimaalisesti. Teoreettisella tasolla tietoliikenneverkkoa voidaan mallintaa graafina  $G$ , joka koostuu solmuista  $V$  ja solmuja toisiinsa liittävästä kaarista  $E$ . Polku on joukko solmuja  $A$  ja  $B$  yhdistäviä solmuja kaaria. Kaaren yli kulkemisen suositussarvon käänteisarvoa kutsutaan kaaren hinnaksi. Graafiteorian kiinnostavin osa reitityksen kannalta on lyhimmän polun ongelma, joka tarkoittaa halvimman mahdollisen polun löytämistä solmujen  $A$  ja  $B$  välillä. Kaksi yleisesti käytössä olevaa ongelman ratkaisevaa algoritmia ovat Bellman-Ford ja Dijkstra, joista ensimmäistä käytetään yleisesti etäisyysvektori-protokollien kanssa ja jälkimmäistä etäisyysvektori-protokollien yhteydessä.

Internet on suuri maailmanlaajuinen tietoliikenneverkko, joka yhdistää useita itsenäisesti reitittyviä autonomisia alueita yhdeksi kokonaisuudeksi. Internetin perusprotokollat on suunniteltu toimimaan ympäristössä, jossa on suhteellisen pienet viiveet ja pakettihäviöt tapahtuvat ruuhkautuvien reitittimien puskurien täyttyessä. Myöskin päästä-päähän yhteys lähettävän ja vastaanottavan solmun välillä vaaditaan. Erityisvälitysverkot ovat tietoliikenneverkkoja, joissa nämä oletukset eivät päde, vaan suuret viiveet ja jatkuvat pakettihäviöt ovat normaalia verkon toimintaa. Esimerkkinä tämän kaltaisista ympäristöistä on planeettojen välisessä viestinnässä käytettävät verkot. DTN on planeettojen välisen viestinnän pohjalta kehitetty alusta, joka mahdollistaa viestinnän erityisvälitysverkkojen kaltaisissa olosuhteissa.

Mobile Ad Hoc Network (MANET)-verkot koostuvat liikkuvista jäsenistä ja niiden välisistä langattomista linkeistä, jotka olettavat päästä-päähän -yhteyden löytyvän verkon solmujen väliltä. Tämä oletus mahdollistaa IP-pohjaisten tekniikoi-

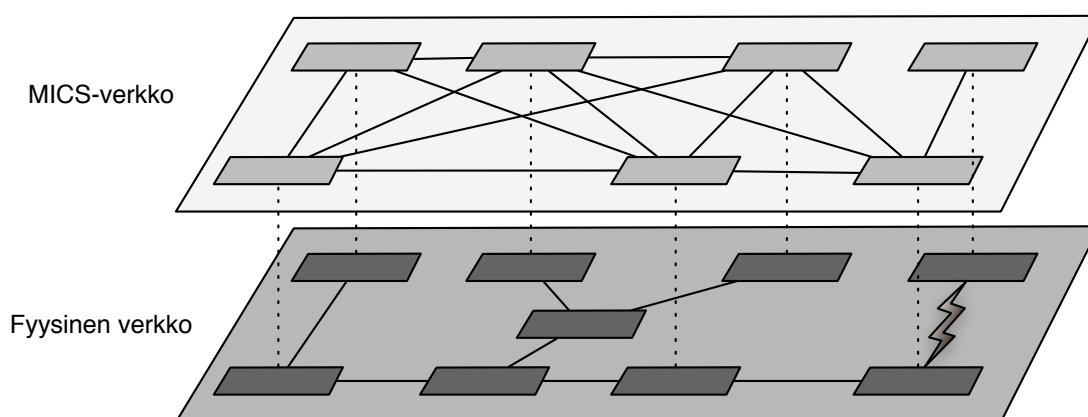
den käyttämisen MANET-verkoissa. MANET-verkoissa käytettyjä reititysprotokollia on esimerkiksi etäisyysvektoriin pohjainen Ad Hoc On-Demand Distance Vector -protokolla ja linkkitilapohjainen Optimized Link State Routing.

Intermittently Connected Mobile Ad Hoc Network (ICMAN) -verkot ovat MANET-verkkoja, joissa ei ole vaatimusta päästä päähän -yhteydestä. Tästä johtuen ICMAN-verkot hyödyntävät yleisesti DTN-verkoista tuttua talleta-kanna-välitä (engl. Store-Carry-Forward, SCF) -menetelmää, jossa välittävät solmut tallentavat välitettävät viestit mahdollisesti pitkäksi ajaksi muistiinsa ja välittävät niitä eteenpäin sopivan solmun tullessa naapurikseen. Tämän kaltaisissa verkoissa käytetään usein tulvitusta, jota voidaan optimoida verkon historiaan perustuvia todennäköisyyksiä käyttäen.

### 3 Dynaaminen reititys MICS-ympäristössä

Tässä luvussa tutustutaan Multi Interface Communication Software (MICS) [26] toimintaan yleisellä tasolla ja perehdytään reitityksen kannalta oleellisiin komponentteihin. Lisäksi käydään läpi teoriatasolla miten dynaaminen reititys toteutettiin järjestelmään. Luvussa 4 tutustutaan dynaamisen reitityksen toteutuksessa käytettyihin yksityiskohtiin.

#### 3.1 Multi Interface Communication Software



Kuva 6: Fyysisen verkon ja MICS-verkon välinen suhde

MICS on haasteellisiin ympäristöihin suunniteltu DTN-järjestelmän kaltainen sanomanvälitysjärjestelmä. Se tarjoaa mahdollisuuden sanomanvälitykseen verkoissa, joissa tiedonvälityksessä yleisesti käytössä olevan TCP/IP-protokollaperheen käyttäminen ei ole mahdollista. Tällaisissa verkoissa on käytössä useita erilaisia siirtoväyliä eikä esimerkiksi päästä-päähän -saavutettavuutta voida taata jatkuvasti. [26] MICS muodostaa fyysisen verkkotopologian päälle oman päällysverkkonsa, joka on esitetty kuvassa 6.

Päällysverkko hyödyntää alla olevaa verkkotekniikkaa muodostaakseen oman topologiansa viestinvälitykseen, joten fyysisesti kaukanakin olevat solmut voivat olla päällysverkon kannalta suoria naapureita. Esimerkiksi yleisesti IP-verkoissa jokainen päätelaite voi muodostaa yhteyden mihin tahansa toiseen päätelaitteeseen useankin hypyn yli. Huomioitavaa kuvassa 6 on se, että oikeassa laidassa radioyhteyden päässä olevalla solmulla (yhteys kuvattu salamalla fyysisellä kerroksella) ei ole yhteyksiä kuin toiseen radiosolmuun. Jotta solmu voi kommunikoida IP-yhteyden kautta saavutettavilla MICS-solmuilla, sen täytyy reitittää liikenne välittäjänä toimivan, sekä

IP- että radioverkkoon kuuluvan MICS-solmun kautta. Tässä työssä tutkitaan ja toteutetaan dynaaminen reititys MICS-järjestelmän muodostamalle päällysverkolle.

## MICS-arkkitehtuuri

Yleisellä tasolla MICS voidaan jakaa kolmeen osa-alueeseen: järjestelmää käyttäviin ohjelmistoihin, MICS-ytimeen ja verkkorajapintoihin. MICS-ydin koostuu useasta erillisestä komponentista, jotka ovat sovellusrajapinta (engl. Application Interface, AI), viestikäsittelijä (engl. Message Handler, MH), reititin (engl. Router), valvoja (engl. Monitor) ja reititysydin (engl. Routing Core, RoCo). Näiden lisäksi työn kannalta oleellinen osa MICS-kokonaisuutta on ilmoittautuja (engl. Affiliation) - apuohjelma, joka hoitaa solmun ilmoittautumisen toiseen solmuun käyttäjän niin pyytäessä.

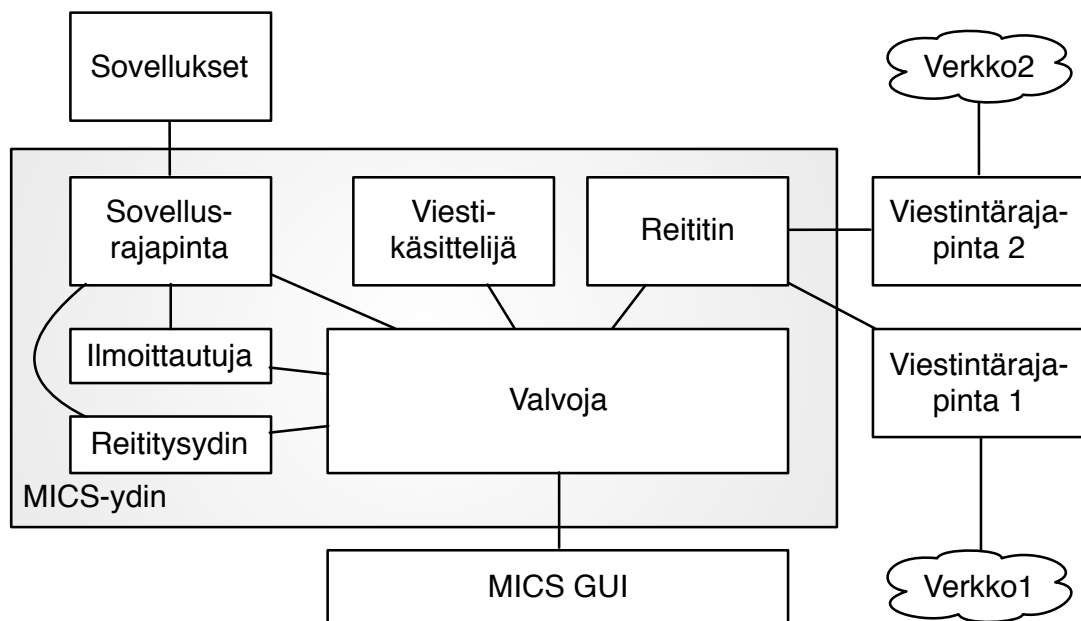
Valvoja toteuttaa prosessien väliseen viestintään (engl. Inter-process Communication, IPC) käytetyn D-Bus-väylän, jonka kautta MICS-ytimen eri osat hoitavat keskenäisen viestintänsä. D-Bus on julkaisija-tilaaja -mallin toteuttava viestiväylä, joka on käytössä keskeisessä asemassa erityisesti Linuxin työpöytäympäristöissä [27].

Sovellusrajapinta tarjoaa käyttäjärajapinnan järjestelmään esittäytymällä sähköpostipalvelimena. Se siirtää viestejä viestikäsittelijän ja käyttäjäohjelmistojen välillä. Viestikäsittelijä hoitaa osoitteiden selvityksen, sanomien muuttamisen natiivimuotoon, pakkaamisen, salaamisen ja päästä-päähän -kuittaukset. Viestinkäsittelijä toimii sovellusrajapinnan ja reitittimen välissä.

Reititin vastaa viestintäraajapintojen ja viestikäsittelijän välisestä liikenteestä, viestien välittämisestä ja viestien lohkomisesta. Reititysydin vastaa MICS-verkon dynaamisesta reitityksestä. Reititysydin lähettää viestinsä solmulta toiselle sähköpostirajapinnan yli ottamalla yhteyden sovellusrajapintaan, ja ilmoittaa reittipäivitykset reitittimelle D-Bus-väylän ylitse. Ilmoittautuja hoitaa MICS-solmun ilmoittautumisen toiseen solmuun, ja vastaavasti myös irtoamisen solmusta. Ilmoittautuminen ja irtoaminen on käyttäjän suorittama manuaalinen toimenpide, jolla voidaan ohjeistaa järjestelmää valitsemaan tietty solmu oletusyhdyksikäytäväksi liikennöintiin muualle verkkoon.

Edellä mainittujen osien lisäksi käytössä on järjestelmän graafinen käyttöliittymä (Graphical User Interface, GUI). Graafinen käyttöliittymä on tarkoitettu pääasiassa ylläpitoon ja MICS-solmujen tilojen tarkkailuun, eikä loppukäyttäjällä ole normaalitilanteessa tarvetta käyttää sitä. Kuvassa 7 esitetään MICS-järjestelmän komponentit ja niiden väliset suhteet graafisesti. Järjestelmän kanssa voidaan käyttää muitakin komponentteja, mutta ne eivät ole työn kannalta oleellisia.

MICS käyttää sanomien välityksessä kaksitasoista osoitteistusta. Käyttäjille nä-



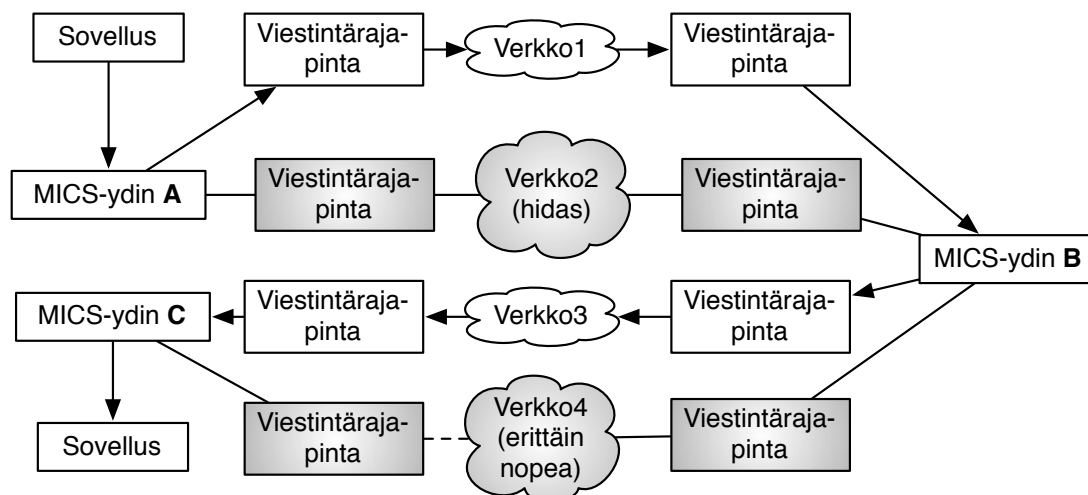
Kuva 7: MICS-arkkitehtuuri

kyvä rajapinta on sähköposti, eli käyttäjän näkökulmasta osoitteistuksena toimii sähköpostiosoite. Sisäisesti järjestelmä käyttää päälysverkkonsa osoitteistuksessa 32-bittistä kokonaislukua, jota kutsutaan MICS tunnisteeksi (engl. MICS ID). Tämä tunniste globaalisti yksilöllinen verkon solmuilla.

Kuvassa 8 esitetään mahdollinen sanomanvälitystilanne. MICS-solmulla A oleva sovellus lähettää sanoman solmulla C olevalle sovellukselle. Solmuilla ei ole suoraa yhteyttä keskenään, vaan liikennöinti tapahtuu solmun B kautta. A- ja B-solmujen välillä on käytössä kaksi eri verkkoa, joista MICS valitsee käyttöön nopeamman verkon 1. Solmujen B ja C välille on myös asettu kaksi erillistä verkkoa, mutta koska solmun C yhteys nopeampaan verkkoon 4 on pois käytöstä, käytetään verkkoa 3 liikennöintiin. Käytöstä pois olevat rajapinnat ja verkot ovat tummennettu kuvaan selvyiden vuoksi. Huomattavaa on, että mikäli solmujen A ja B välinen verkko 1 hajoaa, toimii sanomanvälitys yhä solmujen A ja C välillä verkon 2 kautta. Vastavasti jos verkon 4 kautta toimiva yhteys palaa käyttöön solmujen B ja C välille, hoidetaan sanomanvälitys sen kautta.

### Viestintärajapinnat

Viestintärajapinnat (engl. communication interfaces) ovat erillisiä sovelluksia, jotka toteuttavat liikennöinnin tiettyä verkkoteknologiaa hyväksi käyttäen. Esimerkkejä erilaisista viestintärajapinnoista ovat IP- ja SMS (Short Message Service)-



Kuva 8: Esimerkki viestinvälityksestä MICS-verkossa

rajapinnat: IP-rajapinta voi tarjota erittäin nopean ja luotettavan yhteyden MICS-solmujen välille, mutta SMS-rajapinta on parhaimmillaankin hidas ja rajoittunut siirtoväylä.

Viestintärajapinnat keskustelevat reitittimen kanssa käyttäen NLng-protokollaa [28]. NLng on laajennus Linuxin ytimessä käytettyyn Netlink-prokollaan [29]. Rajapinnat kertovat reitittimelle muun muassa tiedot naapureista, eli kauttaansuoraan saavutettavissa olevista MICS-solmuista. Naapuritietoon kuuluu naapurisolmun MICS tunniste sekä tällä hetkellä tiedossa oleva tila. Kun reititin vastaanottaa tiedon naapurin tilassa tapahtuneesta muutoksesta, signaloi se tiedon muutoksesta D-Bus väylän yli kiinnostuneille komponenteille. Tätä dynaamisen reitityksen toteuttava reititysydin käyttää hyödykseen muodostaessaan omaa linkkitilatietokantansa. Lisäksi viestintärajapinnat kertovat itsestään käytössä olevan tiedonsiirtonopeuden ja viestintäviiveen, joiden perusteella lasketaan hinta rajapinnan takaa löytyville naapureille.

Naapurin tilaa kuvataan Netlink-protokollasta lainatuilla tunnisteilla [29]. Käytössä olevat tilat ovat vajavainen (engl. incomplete), saavutettava (engl. reachable), vanhettunut (engl. stale), tunnusteltava (engl. probe) ja epäonnistunut (engl. failed), jotka esitellään taulukossa 3. Saavutettava kuvaa naapuria, jonka kanssa ollaan liikennöity onnistuneesti lähiaikoina. Kun liikennöinnistä on kulunut hetki, pudotetaan tila vanhettuneeseen. Vanhettunut tarkoittaa naapuria, jonka kanssa ollaan liikennöity onnistuneesti tai jonka kanssa liikennöinnin oletetaan onnistuvan ongelmitta. Näissä kahdessa tilassa olevia naapureita pidetään reitityksellisesti saavutettavissa olevina, ja ne otetaan mukaan reitinlaskentaan. Missä tahansa muussa



Tila	Reititettävissä	Kuvaus
Saavutettava (Reachable)	Kyllä	Naapurille on liikennöity hetki sitten onnistuneesti, yhteys toimii hyvin todennäköisesti
Vanhettunut (Stale)	Kyllä	Liikennöinti naapurille onnistuu melko todennäköisesti
Tunnusteltava (Probe)	Ei	Yhteydenotto naapuriin epäonnistunut
Epäonnistunut (Failed)	Ei	Naapuruuden kanssa on jokin vakava ongelma
Vajavainen (Incomplete)	Ei	Tiedot naapurista ovat puutteelliset

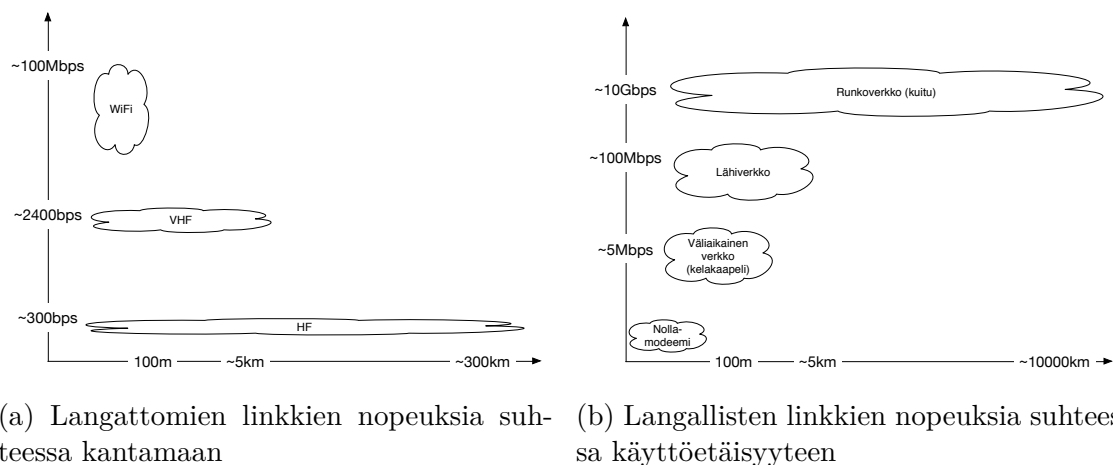
Taulukko 3: Järjestelmän käyttämät naapuritilat

tilassa olevat naapurit eivät ole täysin saavutettavissa, joten niitä ei huomioida reitintalaskennassa.

Kuvassa 9 esitetään erilaisten järjestelmän käyttämien langallisten ja langattomien verkkotekniikoiden nopeutta suhteessa käyttöetäisyyteen. Langattomissa verkoissa (kuva 9a) nähdään selvästi miten kantaman kasvaessa siirtonopeus laskee. Suhteellisen matalien taajuuksien HF-radiot voivat toimia satojen kilometrien etäisyydeltä, mutta siirtonopeus on vain satoja bittejä sekunnissa. Vastaavasti hieman korkeamman taajuuden VHF-radio pääsee muutamaan kilometrin etäisyyteen ja 2400 bps siirtonopeuteen. Langattomat lähiverkot (IEEE 802.11) voivat tekniikasta riippuen päästä jopa yli 100 megabitin sekuntinopeuteen, mutta kantama jää alle 100 metrin luokkaan.

Langallisissa verkoissa (kuva 9b) suhde näyttää menevän juuri toisinpäin. Tämä johtuu siitä, että runkoverkoissa käytetty kuitutekniikka on huomattavan kallista, eivätkä normaalit tietokoneet pysty prosessoimaan tietoa näin korkeilla nopeuksilla. Runkoverkon kuitulinkeillä käytetään tiedonsiirtoon erikoistettua laitteistoa, ja linkit ovat usein esimerkiksi kaupunkien tai jopa valtioiden välisiä maahan kaivettuja yhteyksiä. Vastaavasti hitaita kiinteitä yhteyksiä käytetään erikoistarkoituksiin, esimerkiksi katastrofialueella nopeasti koottu kenttäverkko voi olla hyvinkin hidas pienellä alueella toimiva kiinteä verkko.

Kuten edellä kerrottiin ja kuvasta 9b nähdään, on MICS-verkossa käytössä mahdollisesti erittäin laaja-alaisesti eri nopeuksisia ja erilaisella käyttöperiaatteella toimivia verkkolaitteita. MICS hoitaa viestinnän piilottaen todellisen monimutkaisen ja heterogeenisen verkon käyttäjältä kokonaan.



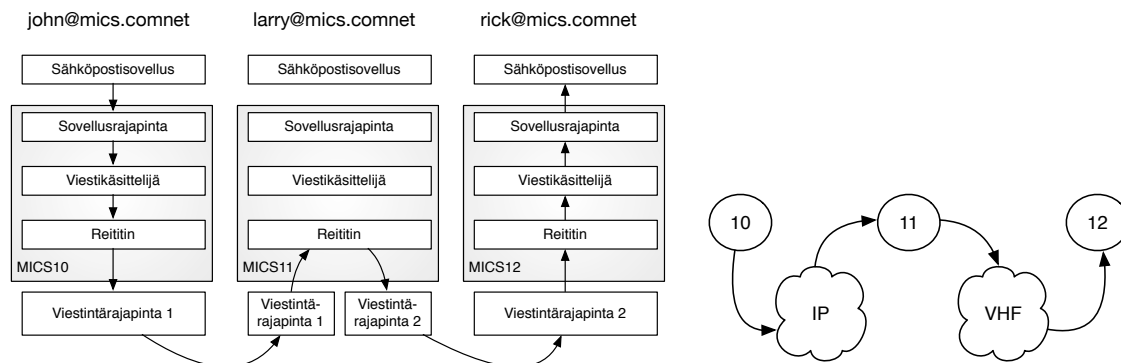
Kuva 9: Langattomien ja langallisten verkkotekniikoiden nopeuksien ja kantamien suuruusluokat

### Sanoman kulku eri komponenttien ja verkon välillä

MICS koostuu useasta edellä mainitusta komponentista, joilla jokaisella on oma tarkkaan määritetty tehtävänsä. Sanoman kulkiessa verkossa kaikki ydinkomponentit osallistuvat sen siirtoon omalla osallaan. Kuvassa 10 esitetään viestin kulku solmulta toiselle. Tässä tapauksessa solmulla 10 sijaitseva käyttäjä john@mics.comnet lähettää käyttämällään sähköpostisovelluksella sanoman käyttäjälle rick@mics.comnet kuvan 10a esittämällä tavalla. Kuva 10b esittää verkkorakenteen hieman korkeammalla tasolla. Solmulla 10 on käytössään vain IP-verkkorajapinta ja vastaanottaja käyttää ainoastaan VHF-verkkorajapintaa, joten sanoman välitys täytyy tehdä solmun 11 kautta jolla on molemmat verkkorajapinnat käytössään. Sanoma lähtee käyttäjältä sovellusrajapinnalle SMTP-protokollaa [30] käyttäen. Sovellusrajapinta edelleenvälittää viestin viestikäsittelijälle.

Viestikäsittelijä jäsentää viestin ja tarkistaa tietokannasta miltä solmulta käyttäjä rick@mics.comnet löytyy. Käyttäjä sijaitsee solmulla 12, joka on eri kuin viestikäsittelijän oma solmu. Omalla solmulla sijaitsevalle käyttäjälle viesti olisi kopioitu vastaanottavan käyttäjän postilaatikkoon, josta sovellusrajapinta olisi sopivalla hetkellä noutanut sen käyttäjälle. Tässä tapauksessa sanoma lähtee toiselle MICS-solmulle, joten Viestikäsittelijä muuttaa sanoman MICS-natiivimuotoon, salaa ja pakkaa sen, ja välittää syntyneen sanoman reitittimelle.

Reititin tarkistaa löytyykö reititystaulusta reittiä kohti sanoman vastaanottajaa solmua 12. Reititystaulusta löytyy merkintä, jonka mukaan kohteelle 12 seuraava hyppy on solmu 11, joka löytyy IP-verkkorajapinnan naapurina. Reititin paloittelee sanoman ja lähettää sen NLog-protokollan yli IP-verkkorajapinnalle. IP-



(a) Käyttäjä john@mics.comnet lähettää sanoman käyttäjälle rick@mics.comnet. (b) Esimerkissä käytetty verkko

Kuva 10: Viestin välittyminen MICS-komponenttien välillä

verkkorajapinta välittää sanoman solmun 11 vastaavalle verkkorajapinnalle, josta sanoma siirtyy reitittimelle.

Solmun 11 reititin tarkistaa jälleen löytyykö reititystaulustaan reittiä kohti solmua 12. Kohde löytyy suorana naapurina VHF-verkkorajapinnan takaa, joten sanoma välitetään sinne. VHF-verkkorajapintojen jälkeen sanoma saapuu kohdesolmun reitittimelle, jossa huomataan sanoman olevan kohdesolmulla. Reititin kokoaa mahdollisesti paloittelun sanoman ja välittää kokonaisen sanoman viestikäsittelijälle, joka poistaa sanomasta pakkauksen ja salauksen, ja tarkistaa löytyykö käyttäjä rick@mics.comnet tältä solmulta. Käyttäjä löytyi, joten sanoma välitetään vastaanottajan sähköpostilaatikkoon, josta käyttäjä voi sen noutaa sovellusrajapinnan kautta IMAP-protokollaa käyttäen [31].

## Ilmoittautuminen

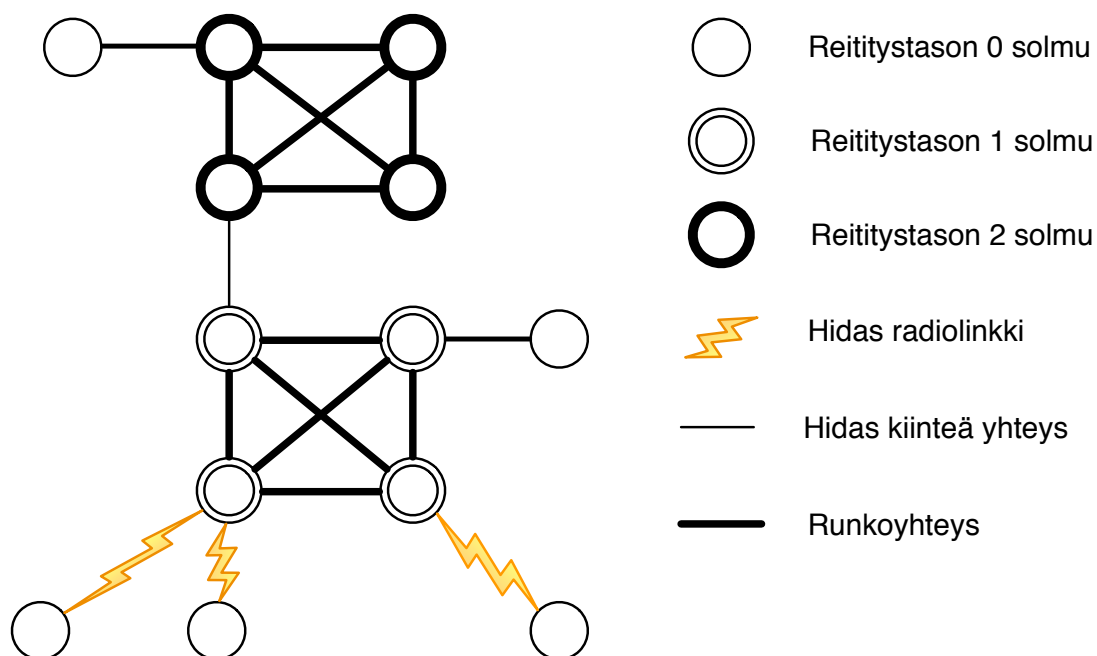
Ilmoittautuminen (engl. affiliation) on käyttäjän suorittama manuaalinen prosessi, missä verkossa liikkuva käyttäjä valitsee minkä solmun kautta haluaa liikennöidä muun MICS-verkon kanssa. Tällä manuaalisella toimenpiteellä nopeutetaan huomattavasti verkon halutun topologian muodostumista esimerkiksi tapauksissa, jossa käyttäjä on useamman tukiaseman kuuluvuusalueella ja vaihtaa radiotukiasemaa. Ilman manuaalista ilmoittautumista kestäisi mahdollisesti pitkään, ennenkuin tieto käyttäjän sijainnista häviäisi vanhalta tukiasemalta. Ilmoittautumisen kanssa saadaan käyttäjän uusi sijainti verkon tietoon jopa sekunneissa. Käyttäjän sijainnin määrittämisen nopeuttamisen lisäksi ilmoittautumisella on muitakin tehtäviä, esimerkiksi haluttujen tilausryhmien uudelleentilaaminen uudelta tukiasemalta. Tilausryhmät eivät ole oleellisia tämän työn kannalta, joten niitä ei käsitellä tässä

yhteydessä.

Käyttäjä voi kerrallaan olla ilmoittauneena vain yhteen MICS-solmuun. Jos käyttäjä ilmoittautuu uuteen solmuun vanhan liittymän ollessa vielä voimassa, lähettää MICS automaattisesti tiedon vanhalle tukiasemalle liikkuvan solmun siirtymisestä uuteen sijaintiin verkossa.

### 3.2 MICS-verkkojen topologiat

Vaikka MICS ei teknisesti rajoita mahdollisia käytettäviä verkkotopologioita, on toimintaa käytännössä optimoitu tietyn muotoisille verkoille. Kuvassa 11 esitetään yksinkertaistettu mallitopologia ja käytössä olevat linkkityypit. Mallitopologiassa olevan yhden solmun voidaan ajatella vastaavan mahdollisesti useaa solmua käytännön verkoissa.



Kuva 11: MICS-verkon esimerkkitopologia

Reititystason 0 solmut ovat niin sanottuja päätesolmuja (tai liikkuvia solmuja), joiden taakse verkkoteknisesti ei ikinä oleteta tulevan muita MICS-solmuja. Tällä tavalla toimimalla voidaan tehdä oletamus, ettei tason 0 solmujen tarvitse ikinä lähettää reititystietoa verkkoon. Myöskään verkon ei tarvitse lähettää reititysviestejä tasolle 0, koska solmut muodostavat automaattisesti oletusyhdyskäytävän korkeampaa reititystasoa kohden, jolloin kaikki liikenne kulkee tukiasemaa kohden. Tällä rajoituksella voidaan mahdollistaa erittäin hitaiden linkkien toimimisen osa-

na MICS-verkkoa dynaamisen reitityksen kanssa ilman, että dynaamisen reitityksen aiheuttama kontrolliliikenne tukkii käytössä olevaa mahdollisesti erittäin kapeaa kaistaa.

Runkoyhteys on verrattaen nopea, mutta ei välttämättä luotettava tai vakaa tiedonsiirtoväylä. Fyysisellä tasolla runkoyhteys voi olla toteutettu lähes millä tahansa tekniikalla radiolinkkien ja valokuidun väliltä. Yhteydet voivat olla heikkolaatuisia, mikä aiheuttaa MICS-tason naapureissa vaihtelua, ja tätä kautta lisää dynaamisen reitityksen kontrolliliikennettä.

Hidas kiinteä yhteys yhdistää usein erillisiä verkkoalueita, ja voi toimia esimerkiksi tietoturvarajapintana. Yhteys on hyvin erilainen verrattuna radiolinkkeihin ja runkoyhteyksiin. Vaikka yhteys onkin hidas, se voi silti olla kymmeniä kertoja radiolinkkiä nopeampi, ja toisaalta se kiinteänä yhteytenä saattaa olla vakaampi kuin runkoverkon yhteydet. Koska nämä linkit ovat kriittisiä verkon viestinnän kannalta, saattaa niitä olla verkossa toiminnassa useampi eri solmujen välillä virhesietoisuuden parantamiseksi.

Edellä mainitusti mallitopologiassa sijaitseva solmu voi vastata useaa solmua käytännön verkoissa. Tätäkin isompia verkkoja voidaan muodostaa asettamalla useita kuvassa 11 esitetyn tyyppisiä verkkoja rinnakkain ja yhdistämällä ne toisiinsa korkeammalla reititystasolla.

### 3.3 Käytetty linkkitilareititysprotokolla

Intermediate System to Intermediate System (IS-IS) on ISO/IEC 10589 -standardissa [7] määritelty linkkitilan levitykseen perustuva reititysprotokolla. Toisin kuin esimerkiksi OSPF, se ei ole sidottu tiettyyn siirtokerroksen protokollaan, ja sen Type-Length-Value (TLV) -malli takaa laajennettavuuden tarpeen mukaan. Näiden ominaisuuksien perusteella IS-IS valittiin MICS-järjestelmän dynaamisen reitityksen perustaksi. Tässä aliluvussa käydään läpi MICS-järjestelmän kannalta oleellisia IS-IS-protokollan perusteita.

#### Perusteet

IS-IS on perinteinen linkkitilaprotokolla, jonka toiminta perustuu naapurien kokoamiseen linkkitilaviestiin, joka tulvitetaan verkon muille reitittimille. Näiden linkkitilaviestien perusteella saadaan muodostettua verkon rakenne, josta lasketaan lyhimät mahdolliset reitit jokaiselle muulle tunnetulle kohteelle.

Standardi IS-IS protokolla jakautuu kahteen aliprotokollaan. Hello-protokollaa käytetään naapureiden tunnistamiseen ja tilan tarkkailuun, joiden perusteella luo-

daan naapuritietokanta. Linkkitilaprotokollaa käytetään naapuritietokannan sisällön jakamiseen verkkoon. MICS-järjestelmässä varsinaiselle Hello-protokollalle ei ole tarvetta, koska viestintäraajapinnat hoitavat naapurien hankkimisen ja tilan päivittämisen joka tapauksessa.

## Reititystasot

IS-IS käyttää skaalautuvuuden parantamiseksi kahden tason mallia. Tason 1 reititimet hoitavat reitityksen saman alueen sisällä, ja jakavat tiedon alueen sisäisestä reitityksestä ylös runkoverkon tason 2 reitittimille, jotka jakavat keskenään tiedot kaikista tason 1 alueista. Jos tason 1 reititin ei löydä kohdetta omasta reititystaulustaan, lähetetään paketti tasolle 2. Jos kohde on olemassa, osaa tason 2 reititin ohjata paketin oikeaan suuntaan.

## Reititysalueet

Standardin IS-IS-protokollan alueita (engl. area) käytetään isojen verkkojen jakamiseen pienemmiksi kokonaisuuksiksi. Samaan alueeseen kuuluvat reitittimet muodostavat oman reititysalueensa (engl. routing domain). Tasolla 1 reitittimien naapuruudet syntyvät vain, jos reitittimet kuuluvat samaan alueeseen. Yksi reititin voi kuulua kerralla maksimissaan kolmeen eri alueeseen.

Reititys eri alueiden välillä tapahtuu runkoverkon (reititystason 2) reitittimien kautta.

## TLV-rakenne

IS-IS on helposti laajennettavissa oleva protokolla. Laajennettavuus perustuu joustavaan Tyyppi-Pituus-Arvo (Type-Length-Value, TLV) -rakenteeseen. Esimerkki TLV-rakenteesta on esitetty kuvassa 12.

Koko (tavua)	
TLV:n tyyppi	1
TLV:n pituus	1
Data	1-255

Kuva 12: TLV-objektien perusrakenne

Sanomissa on helppo ketjuttaa TLV-rakenteita peräkkäin lähes rajattomasti. Lisäksi näiden jäsentäminen (engl. parsing) ja sarjallistaminen (engl. serializing) on

yksinkertaista. Koska TLV-rakenteessa tyyppiä kuvaa vain yksi tavu, on erilaisia TLV-tyyppejä mahdollista olla vain 255 erilaista. Koska yleisessä tapauksessa tämä ei ole riittävästi, on otettu käyttöön ali-TLV (engl. sub-TLV) käsite. Ali-TLV on tavallisen TLV-objektin sisällä olevan toinen TLV-objekti. Tällä mallilla saadaan laajennettua käytössä olevaa nimiavaruutta huomattavasti.

## Viestityypit

IS-IS käyttää viestinnässään kolmea erilaista viestityyppiä: Hello-viestejä, Link State Protocol Data Unit (LS PDU, LSP) eli linkkilaviestejä ja sekvenssinumeroviestejä. Huomattavaa on, kuten aliluvun alussa mainittiin, että MICS-järjestelmässä ei ole tarvetta Hello-viesteille.

Kaikki edellä mainitut viestit alkavat yhteisellä otsikolla (engl. header), joka esitetään kuvassa 13.

Kentän nimi				Vakioarvo	Koko (tavua)
Tunniste				0x83	1
Otsakkeen pituus					1
Versio				1	1
ID:n pituus				6 (0)	1
V	V	V	PDU:n tyyppi		1
PDU:n versio				1	1
Varattu				0	1
Alueiden sallittu määrä				3 (0)	1
Viestin tyyppille ominaiset kentät					17-33
TLV-osio					0-1467

Kuva 13: IS-IS-protokollan kahdeksan tavun pituinen viestityyppien yhteinen otsikko

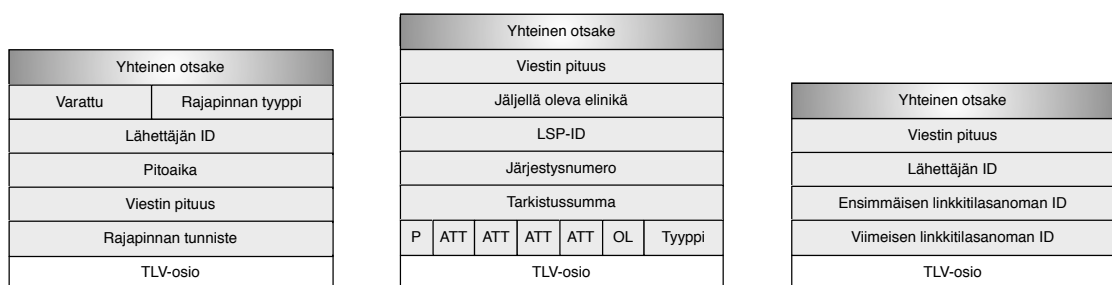
Seuraavaksi esitellään kokonaisuuden ymmärtämisen ja työn kannalta oleelliset IS-IS viestityypit. IS-IS-protokollan käyttämät perusviestityypit löytyvät IS-IS-standardista [7].

## Hello-sanomat

Naapuruudet muodostetaan käyttämällä Hello-sanomia. Jokainen reititin lähettää säännöllisin väliajoin Hello-sanoman, johon on liitettynä tieto kenen kaikkien muiden reitittimien Hello-sanoman viestin lähettäjä on nähnyt linkillä. Kun reititin vastaanottaa kyseisen sanoman, jossa ilmoitetaan, että toinen reititin on nähnyt

vastaanottavan reitittimen Hello-sanoman, lisää vastaanottava reititin toisen reitittimen tiedot omaan naapuritietokantaansa. Mikäli naapuritietokannasta löytyvän reitittimen Hello-sanomia ei vastaanoteta tarpeeksi pitkään aikaan, oletetaan yhteyden reitittimien välillä katkenneeksi ja poistetaan merkintä toisesta reitittimestä naapuritietokannasta. Tyypillinen aikaväli Hello-sanomien lähetykselle on 10 sekuntia, ja 30 sekunnin jälkeen ilman sanomaa naapuri poistetaan naapuritietokannasta. Nämä ovat oletusasetukset Ciscon IOS-käyttöjärjestelmässä [32].

Hello-sanomia on kahta eri tyyppiä: monilähetyslinkeillä (engl. broadcast link) käytettäviä yhdellä lähetyksellä usealla vastaanottajalle päätyviä sanomia, ja vastaavasti suoraan kahden reitittimen välisillä Point-to-Point -tyyppisillä linkeillä käytettäviä sanomia. Monilähetyslinkeillä on käytössä oma viestityyppi jotta linkille lähetettävää viestimäärää saadaan vähennettyä. Yksi linkillä oleva reititin äänestetään linkistä vastuussa olevaksi reitittimeksi (Designated Intermediate System, DIS), joka esiintyy linkillä näennäissolmuna (engl. pseudonode). Sen sijaan että samalla linkillä olevat reitittimet lähettäisivät kaikki uuden linkkitilasanoman linkille muutoksen tapahtuessa, ainoastaan näennäissolmu lähettää sanoman jaetulle linkille. Tällä tavalla toimimalla saadaan linkille lähetettävä viestimäärä tiputettua suuruusluokasta  $O(N^2)$  luokkaan  $O(N)$ . Kuvassa 14a esitetään point-to-point -linkeillä käytetyn Hello-viestin otsikon rakenne.



(a) Hello-sanoman otsikko P2P-linkeillä (b) Linkkitilasanomien otsikko (c) CSNP-sanoman otsikko

Kuva 14: Yleisien IS-IS sanomien otsikot

Koska IS-IS-protokollan ja muiden IGP-reititysprotokollien Hello-protokollien ajastimet ovat melko pitkiä, kestää linkkihäiriöiden havaitsemisessa mahdollisesti pitkäänkin. Nykyisissä suurinopeuksisissa linkeissä jo sekunninkin katko aiheuttaa suuren pakettihäviön. Tästä syystä monissa verkoissa on käytössä nykyään Bidirectional Forwarding Detection (BFD) [33] -protokolla IGP-protokollien omien Hello-protokollien tukena. BFD on pohjimmiltaan hyvin yksinkertaistettu Hello-protokolla jonka tehtävänä on havaita häiriöt linkillä mahdollisimman nopeasti, jopa millise-



kuntiluokassa asetuksista riippuen. BFD ei tarjoa esimerkiksi naapurien etsintää, joten se toimii vain lisätietolähteenä IGP-protokollille. Myös esimerkiksi Multiprotocol Label Switching (MPLS) voi hyödyntää sitä [34] [35].

### Linkkitilaviestit

Naapuritietokannan perusteella luodaan niin sanottu linkkitilaviesti, jossa on listattuna kaikki naapuritietokannasta löytyvät reitittimet. Näitä viestejä tulvitetaan koko verkossa, kunnes kaikki verkon reitittimet ovat vastaanottaneet linkkitilaviestin kaikilta muilta reitittimiltä. Tämän linkkitilatietokannan perusteella reititin voi laskea halvimman mahdollisimman reitin kaikille muille reitittimille koko verkossa. Kuvassa 14b esitetään linkkitilaviesteissä käytetyn otsikon rakenne.

### Sekvenssinumeroviestit

Sekvenssinumeroviestejä käytetään linkkitilatietokantojen synkronoimiseen ja luotettavuuden luomiseen. Täydellinen sekvenssinumeroviesti (Complete Sequence Numbers PDU, CSNP) sisältää tiedon kaikista linkkitilatietokannasta löytyvistä LSP-viesteistä. Osittaista sekvenssinumeroviestiä (Partial Sequence Numbers PDU, PSNP) käytetään kuittausviestinä LSP-viestin vastaanottamisesta. Jos reititin vastaanottaa naapuriltaan täydellisen sekvenssinumeroviestin jossa ei listata reitittimen omasta tietokannasta löytyvää linkkitilaviestiä, lähettää reititin sen naapurilleen. Koska IS-IS toimii verkossa epäluotettavalla tasolla 2, lähetetään sama sanoma tarvittaessa säännöllisen väliajoin uudestaan samalle vastaanottajalle, kunnes vastaanottaja lähettää osittaisen sekvenssinumeroviestin jossa kuitataan kyseinen sanoma vastaanotetuksi. Kuvassa 14c esitetään täydellisen sekvenssinumeroviestin käyttämän otsikon rakenne.

## 3.4 Dynaaminen reititys MICS-verkossa

Järjestelmän ainutlaatuisesta toimintamallista johtuen standardoitua versiota IS-IS-reititysprotokollasta ei voida suoraan käyttää, vaan protokolla vaatii sovittamista erilaiseen ympäristöön. Suurin muutos on varsinaisen Hello-protokollan puuttuminen, koska viestintärajapinnat tarjoavat naapuritiedon dynaamisen reitityksen toteuttavan komponentin, reititysytimen, käyttöön.

MICS myös tarjoaa naapuritiedon lisäksi muita hyödyllisiä ominaisuuksia mitä tavallisesti IS-IS-protokollalla ei ole käytössä, joista luotettava tiedonsiirto on ehkä hyödyllisin ominaisuus. Standardi IS-IS käyttää sarjanumeroviestejä (Sequence Number PDU) varmistamaan, että vastaanottaja on saanut lähetyn viestin. MICS

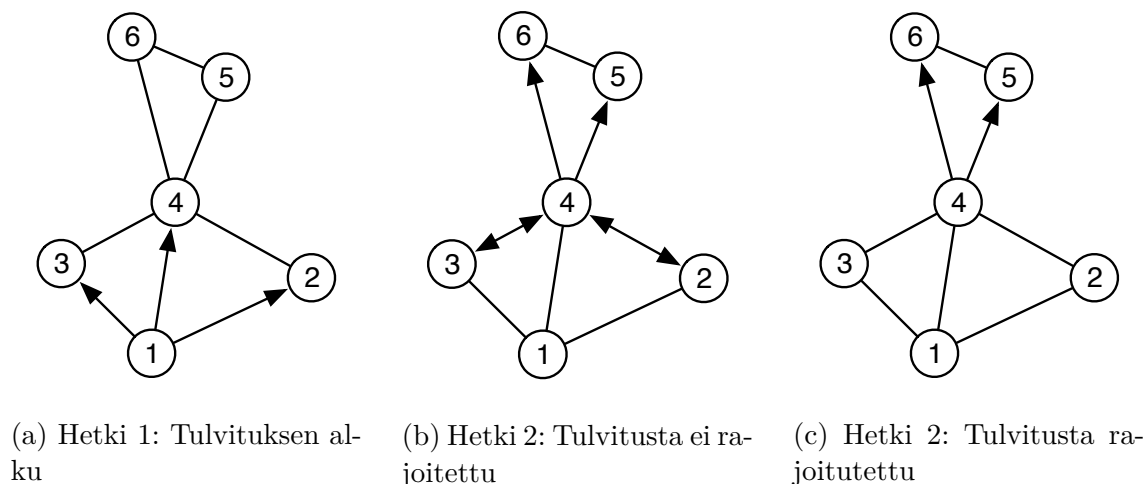
tarjoaa suoraan luotettavaa tiedonsiirtoa, joten lisävarmistuksille ei ole tarvetta reititysytimen käyttämällä tasolla.

### Linkkitilasanomien tulvitus verkossa

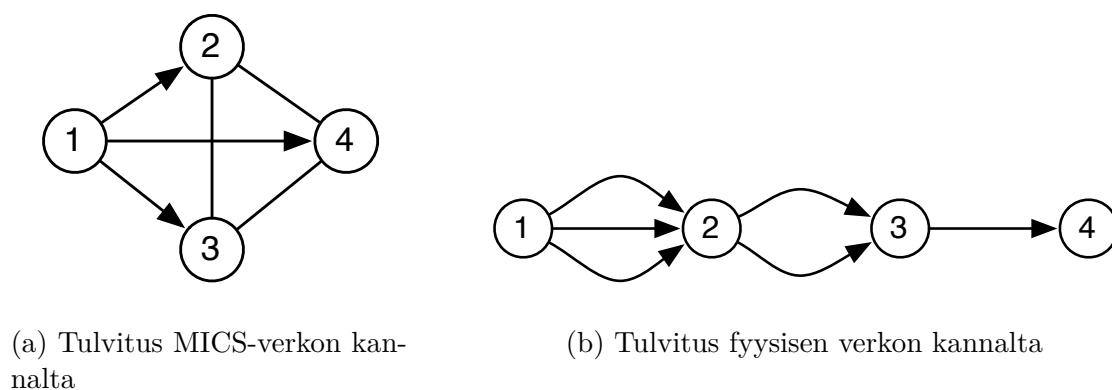
Linkkitilasanomien tulvitus verkossa perustuu yksinkertaiseen ideaan: kun vastaanotat viestin, lähetä se kaikille muille naapureillesi, paitsi alkuperäiselle lähettäjälle. Tämä malli toimii hyvin yksinkertaisissa verkkomalleissa, mutta yleisesti käytössä olevassa täysin kytketyssä (engl. full mesh) verkossa aiheutuu kohtuuttoman paljon turhia uudelleenlähetystyksiä. Jos verkossa on  $N$  solmua, ja jokainen lähettää sanoman kaikille paitsi alkuperäiselle lähettäjälle, tulee verkkoon  $(N - 1) \cdot (N - 2)$  lähetystä. Tätä voidaan optimoida käyttämällä hyödyksi naapuritietoa mitä vastaanottajalla viestin lähettäjältä mahdollisesti on. Jos on tiedossa, että alkuperäisellä lähettäjällä ja vastaanottajalla on yhteisiä naapureita, ei viestin vastaanottajan tarvitse edelleenlähetää yhteisille naapureille sanomaa, koska alkuperäinen lähettäjä on jo tulvittanut viestin omille naapureilleen. Kun verkkotopologia on tiedossa, tämä vähentää äskeisessä esimerkissä tarvittavien viestien lähetysmäärää  $N-1:n$  kappaleeseen.

Kuvassa 15 on esimerkki viestin tulvittamisesta verkossa ilman tulvituksen rajoittamista ja tulvituksen rajoittamisen kanssa. Kuvassa 15a alkuperäinen lähettäjä lähettää sanoman kolmelle naapurilleen. Kuvassa 15b nähdään lähetykset, jotka tapahtuvat ilman tulvituksen rajoittamista: lähetykset, joissa solmut 2 ja 3 ovat osallisina vastaanottajina tai lähettäjinä ovat turhia, koska kaikilla niihin lähetykseen liittyvillä solmuilla on jo kyseinen versio vastaanotettuna. Kohdassa 15c on tulvituksen esto käytössä. Tässä tapauksessa viestin vastaanottanut solmu 4 huomaa, että alkuperäinen lähettäjä on jo lähettänyt viestin naapureilleen solmuille 2 ja 3, joten solmu 4 lähettää viestin eteenpäin ainoastaan solmuille 5 ja 6. Vastaavasti solmut 2 ja 3 huomaavat, että solmulle 4 on jo lähetetty tämä sanoma, joten uudelleenlähetystyksiä ei suoriteta. Tässä esimerkissä tarvittavien viestilähetysten määrä tippui kuudesta kahteen. Kuvassa on esitetty ainoastaan tulvituksen kaksi ensimmäistä vaihetta, kolmannessa vaiheessa solmut 5 ja 6 lähettäisivät viestin toisilleen riippumatta siitä, onko tulvituksen esto käytössä vai ei.

Sanomien tulvitus päällysverkossa ei ole ongelmaton. Kuvassa 16 on esitetty tilanne, jossa päällysverkon kannalta verkko on tähtitopologiassa, mutta fyysisesti koneet on kytketty peräkkäin. MICS tulvittaa naapureilleen eli muille verkon solmuille saman sanoman, jolloin fyysisessä verkossa sama viesti lähetetään ensimmäisen hypyn yli kolme kertaa ja toisenkin hypyn yli kaksi kertaa. Jos MICS-verkko olisi konfiguroitu vastaavaksi kuin fyysinen verkko, tulisi jokaisen fyysisen linkin yli vain



Kuva 15: Sanomien tulvitus verkossa



Kuva 16: Sanomien tulvitus verkossa päällysverkon ja fyysisen verkon kannalta

yksi lähetys. Tämä kuormittaisi mahdollisesti hyvin hitaita linkkejä huomattavasti vähemmän.

### Linkkitilatietokantojen synkronointi

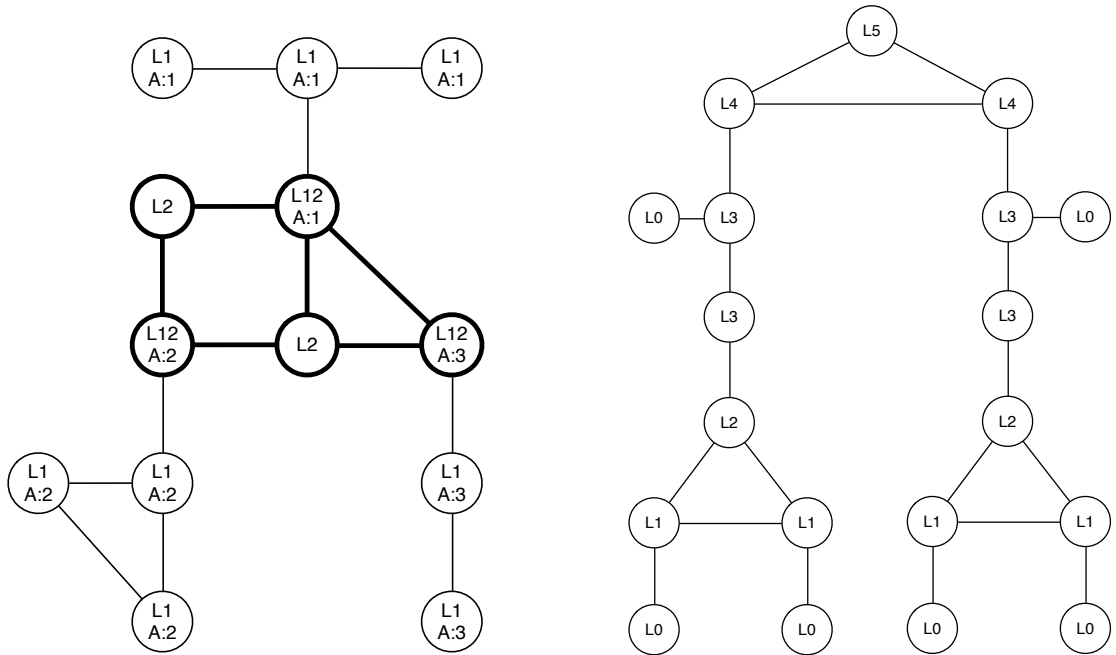
Kun naapuruus syntyy kahden solmun välille, on solmujen synkronoitava linkkitilatietokantansa. Tähän käytetään Complete Sequence Numbers PDU (CSNP) -viestiä. CSNP sisältää listauksen kaikista linkkitilaviesteistä, jotka solmun linkkitilatietokannasta löytyy. Mikäli vastaanottava solmu huomaa omassa tietokannassaan linkkitilaviestin jota toinen puoli ei maininnut omastaan, lähettää CSNP-sanoman vastaanottanut solmu listasta puuttuneen linkkitilaviestin CSNP-sanoman lähettäjälle. Kun molemmat osapuolet lähettävät sanomat toisilleen naapuruuden syntyessä saadaan linkkitilatietokannat synkronoitua. Tarvittaessa uudet linkkitilasanomat

vastaanottanut solmu tulvittaa sanomia verkossa eteenpäin.

Synkronoinnin lisäksi CSNP-sanomat sisältävät tiedot solmun reititystasosta ja -alueesta. Näin saadaan varmistettua että tarvittavat solmukohtaiset tiedot levitetään solmun lähiympäristöön mahdollisimman aikaisessa vaiheessa.

## Reititystasot

Reititysydin laajentaa IS-IS-protokollan kahden reititystason mallia n-tasoiseksi. Useampi reititystaso mahdollistaa aidosti hierarkisen verkon rakentamisen ja reititystiedon levittämisen rajaamisen ainoastaan tarpeellisiin solmuihin. Kuvassa 17a esitetään IS-IS-protokollan käyttämä malli, jossa useampi reititystason 1 reititysalue yhdistetään runkona toimivan reititystason 2 verkon kanssa. Tason 12 solmut ovat reititystasojen 1 ja 2 välillä toimivia, molempiin tasoihin kuuluvia solmuja, joiden kautta tasot liikennöivät keskenään. Keskeinen asia tässä mallissa on se, että reititystason 2 runkoverkon pitää olla yhtenäinen jotta reititys eri alueiden välillä toimii. Käytetyssä olevan ympäristön huomattavasti vapaammasta verkkorakenteesta johtuen yhtenäisen rungon takaaminen ei ole mahdollista, joten mallia on sovellettu MICS-ympäristöön sopivammaksi.



(a) IS-IS -protokollan käyttämä reititystasomalli

(b) MICS-järjestelmän reitityksessä käyttämä reititystasomalli

Kuva 17: IS-IS-protokollan ja MICS-järjestelmän käyttämät reititystasomallit

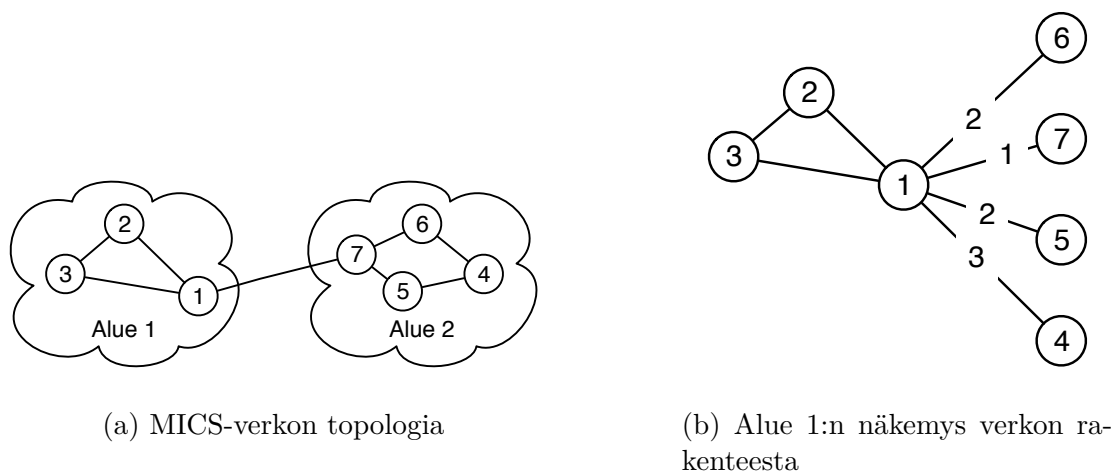
Kuvassa 17b esitetään MICS-ympäristössä käytettävän reitityksen tasomalli. Yhtä runkoverkkoa ei ole, vaan reititystasot muodostavat puumaisen rakenteen verkosta. Solmut levittävät verkossa reititystietoa ainoastaan naapureilleen, jotka ovat samaa tai korkeampaa reititystasoa. Tällä tavoin saadaan laskettua verkossa kulkevan reititysviestinnän määrää huomattavasti.

Poikkeustapauksena reititystasojen kanssa on taso 0. Reititystasolla 0 toimivat solmut, joiden taakse ei MICS-tasolla tule ikinä muita solmuja. Tämä mahdollistaa sen, ettei tason 0 solmujen tarvitse ikinä lähettää linkkilaviestejä verkon suuntaan, eikä ylemmille reititystasojen kuulumien naapuriin tarvitse lähettää viestejä tason 0 solmuille. Kuitenkaan ylemmän tason solmut eivät mainosta tason 0 solmuja verkkoon suoraan naapuruuden syntyessä, vaan käyttäjän pitää suorittaa ilmoittautuminen haluamaan solmuun käsin. Kun ilmoittautuminen on suoritettu, mainostetaan tason 0 solmua verkkoon kuten muitakin naapureita, kunnes solmu ilmoittaa irtautuvansa, kunnes toinen tukiasema ilmoittaa solmun ilmoittautuneen toisalle verkkoon, tai kunnes solmun naapuruus tukiaseman kanssa häviää.

Jotta alemman tason solmut pystyvät kommunikoimaan verkossa ylemmän tason solmujen kanssa, asettavat ne reitityshinnalta halvimman korkeamman tason solmun itselleen oletusyhdyksykäytäväksi muualle verkkoon. Tästä seuraa rajoite: verkko täytyy rakentaa siten, että ylöspäin kulkiessa ei tule haarautumia useisiin kohteisiin. Toisin sanoen, verkossa ylimmällä tasolla olevien solmujen täytyy tietää seuraava hyppy kohti kaikkia verkosta löytyviä solmuja. Jos tietyllä ajanhetkellä reititystietoa verkon ylimmällä tasolla ei ole, ei sanomaa kuitenkaan tiputeta, vaan SCF-mallin mukaisesti se jää odottamaan eliniästään riippuvaksi ajaksi reitin muodostumista.

## Reititysalueet

Reititysalueet ovat toinen tapa rajoittaa linkkilaviestien määrää suurissa verkoissa. Eri reititysalueille kuuluvat solmut eivät vaihda keskenään linkkilaviestejä, vaan ne lähettävät toisilleen säännöllisin väliajoin oman reititystaulunsa sisällön, ja mainostavat oman alueensa sisällä tietävänsä reitin toisen alueen puolelta ilmoitetuille kohteille. Kuvassa 18a on esimerkki reititysalueista. Kaksi reititysaluetta, 1 ja 2, on yhdistetty solmujen 1 ja 7 kautta. Solmut alkavat vaihtaa säännöllisin väliajoin listaa kaikista tuntemistaan solmuista ja kustannuksista niihin, eli alueet yhdistävät solmut vaihtavat reititystaulujaan. Alueiden rajasolmut esittävät toisella alueella sijaitsevat solmut omalle alueelleen omina naapureinaan, joiden kustannus on solmujen saavuttamiseksi vaadittava kustannus. Alueiden rajasolmut kertovat siis omalle alueelleen mitkä solmut ovat saavutettavissa kauttaan. Kuvassa 18b esitetään miten toisella alueella sijaitsevat solmut esitetään omalle alueelle sijaitseville solmuille.



Kuva 18: Esimerkkikuva MICS-järjestelmän reititysalueista

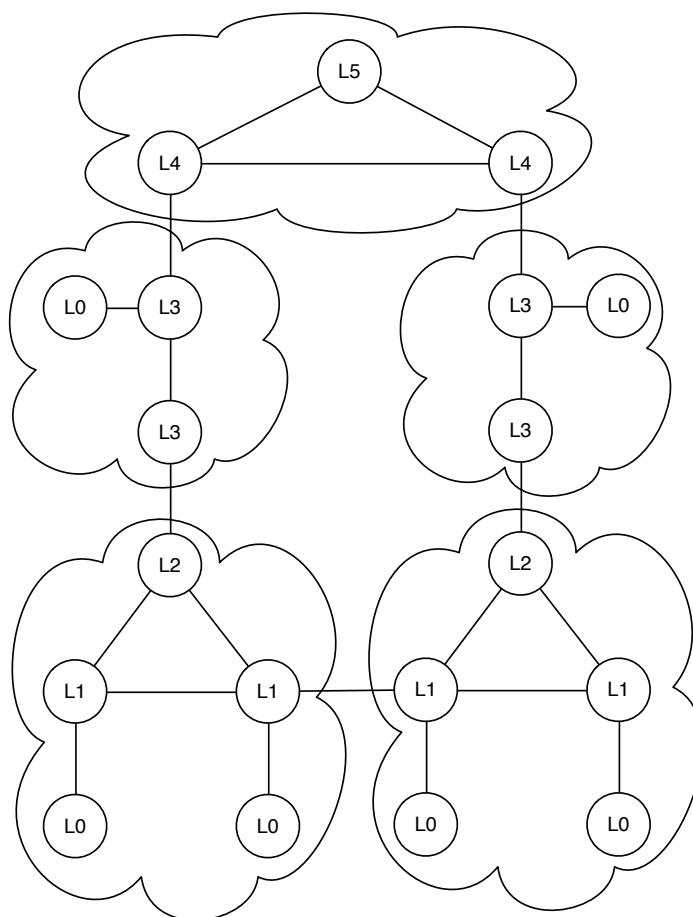
### Reititystasot ja -alueet yhdessä

Jotta dynaaminen reititys saadaan toimimaan tehokkaasti isoissa MICS-verkoissa, täytyy reititystasoja käyttää yhdessä reititysalueiden kanssa. Kuvassa 19 on esitetty esimerkki kuvan 17b verkosta muokattuna reititysalueiden kanssa. Tässä mallissa oletusyhdykäytävä muodostuu reititystasojen mukaan ylöspäin verkossa kuten pelkkien reititystasojenkin kanssa, mutta alueiden välillä lähetetään oletusasetuksilla vain yksi viesti minuutissa. Tämä vähentää huomattavasti liikenteen määrää verkossa, mutta hidastaa verkon tilanteiden päivittymistä koko verkkoon. Verkon laadusta ja solmujen määrästä riippuen tämä on kuitenkin tehtävä verkon toimivuuden säilyttämiseksi.

Topologiassa erona alkuperäiseen verkkoon on uusi yhteys puun haarojen välillä. Ilman tätä yhteyttä liikenne haarojen välillä joutuisi kiertämään puussa kohtuuttoman pitkän matkaa aivan ylhäältä asti, vaikka fyysisesti yhteys olisi mahdollista järjestää alemmalla tasolla. Jos yhteys haarojen välille olisi tehty ilman reititysalueita, leviäisi linkkitilatieta haarojen välillä ja aiheuttaisi kohtuuttoman paljon ylimääräistä liikennettä ja reitinlaskentaa molemmille haaroille. Alueita käyttämällä liikennemäärä pysyy kohtuullisena, ja jos esimerkiksi toisen haaran yhteys ylös verkkoon katkeaa, löytävät sanomat uuden reitin toista haaraa pitkin aivan kuin täysi linkkitilatieta olisi levitetty haarojen välillä.

### Ilmoittaminen verkkoon reitityksen kannalta

Kuten reititystasoista kertovassa kappaleessa kuvattiin, ei reititystason 0 solmuja mainosteta verkkoon naapureina ennenkuin ne ovat ilmoittautuneet haluamaan-



Kuva 19: Esimerkki reititystasoista- ja alueista yhdessä

sa korkeamman reititystason solmuun. Vastaavasti irtoamisen yhteydessä naapuruuden mainostaminen verkkoon lopetetaan, vaikka yhteys solmujen välillä olisikin kunnossa. Syynä tähän toimintaan ovat mahdollisesti hyvin hitaat viestintärajapinnat, joilla naapuruuksien päivittyminen ajantasalle saattaa kestää hyvinkin pitkään, tai joille naapuruus on hyvin abstrakti käsite. Kun käyttäjä haluaa vaihtaa solmua jonka kautta liikennöi verkkoon, riittää toimivan liikennöinnin saavuttamiseen naapuruuden syntymisen jälkeen suoritettu ilmoittautuminen tukiasemaan. Ilman ilmoittautumista voisi kulua pitkäänkin ennenkuin naapuruus häviäisi alkuperäiseltä solmulta.

Jos käyttäjä ei suorita irrottautumista vanhalta solmulta ennen ilmoittautumista uuteen solmuun, lähettää MICS automaattisesti ilmoittautumisen yhteydessä tiedon mille vanhoille tukiasemille käyttäjä on ollut ilmoittautuneena. Liitynnän vastaanotettava solmu lähettää sanoman vastaanotettuaan tiedon vanhalle solmulle käyttäjän siirtymisestä.

### 3.5 Vertailu DTLSR-prokollan kanssa

MICS ja Kevin Fallin esittelemä DTN ovat hyvin samankaltaisia keskenään: molemmat muodostavat päälysverkon jossa toimivat, molemmat käyttävät sähköpostia käyttäjärajapintana, ja molemmat hyödyntävät viestin välityksessä talleta-kannavälitä -menetelmää. Isoin ero järjestelmien välillä onkin reitityksessä. Vaikkakin tiettyihin rajatun tapauksen DTN-verkkojen käyttötapauksiin on esitetty käytettäväksi linkkitilaprotokollaa (DTLSR [24]), käyttää suurin osa DTN-verkoista luonteestaan johtuen stokastisia reititysmenetelmiä. Stokastiset reititysmenetelmät joko kuormittavat verkkoa ylenmäärin tai vaihtoehtoisesti sanomien välittämisessä kestää hyvin pitkään.

DTLSR on suunniteltu käytettäväksi kehittyvillä alueilla missä jatkuvaa internet-yhteyttä ei ole saatavilla, mutta verkko on muuten suhteellisen vakaa. Esimerkki tästä voisi olla syrjäinen kylä, jossa internet-yhteys ulkomaailmaan muodostetaan kylään kaksi kertaa vuorokaudessa saapuvan linja-auton kautta. Vaikka yhteys muualle onkin poikki suuren osan ajasta, on linkin syntyminen hyvin ennustettavissa linja-auton aikataulujen mukaan.

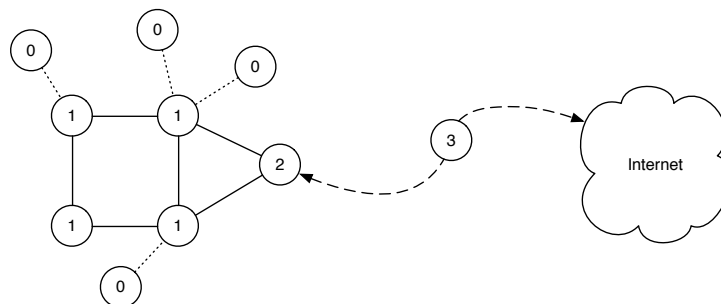
Tässä työssä suunniteltiin ja toteutettiin DTLSR-protokollan tavoin linkkitilaprotokollaan pohjautuva reititys haasteellisiin verkkoihin, mutta varsin erilaisesta käyttöympäristöstä johtuen lopputulos on linkkitilaprotokollan perustoimintaa lukuunottamatta erilainen. Kuten luvussa 2.7 kerrottiin, DTLSR muuttaa linkkitilaprotokollan toimintaa kahdella merkittävällä tavalla: se kasvattaa linkkitilaviestin elinikää huomattavasti, ja ottaa huomioon reitinlaskennassa myös tällä hetkellä alhaalla olevat linkit olettaen, että ne nousevat pian uudestaan. Dynaaminen reititys MICS-järjestelmän yhteydessä ei muuta näitä asioita, mutta hyödyntää yleisesti käytössä olevaa verkkohierarkiaa ja tekee lisäyksiä perustoiminnallisuuteen sen perusteella. Ensimmäiseksi oletamme, että verkon kaikista hitaimmat yhteydet sijaitsevat aivan verkon laidalla, jolloin linkkitilan levittäminen hitaiden yhteyksien yli on tarpeetonta (vaikkakin tarvittassa mahdollista). Toinen oletus on hierarkinen verkotopologia. Sen perusteella voimme olettaa reitin tuntemattomalle solmulle löytävän ylempää hierarkiasta. Tällöin koko verkkoon ei tarvitse turhaan levittää koko linkkitilaa.

Vaikka DTLSR ja dynaaminen reititys MICS-ympäristössä onkin suunniteltu eri lähtökohdista, soveltuu tässä työssä esitetty malli hyvin myös syrjäisen kylän tilanteeseen mihin DTLSR on suunniteltu. Kuvassa 20 esitetään malli syrjäisen kylän omasta verkosta ja sen yhteydestä internetiin MICS-verkkoa hyödyntämällä. Kuvassa olevien solmujen sisällä olevat numerot kuvaavat solmujen reititystasoja. Tason 0 solmut voisivat olla kylän asukkaiden kotona olevia päätteitä jotka on yhdistet-



ty kylän kiinteään verkkoon langattomasti esimerkiksi WiMAX-tekniikalla. Tason 1 solmut kuvaavat WiMAX-tukiasemia, jotka samalla toimivat MICS-solmuina. Reititystason 2 solmu toimii sanomien väliaikaisvarastona ennen kuin ne välitetään internetiin kylän ja kaupungin välillä liikkuvan linja-auton mukana kulkevassa MICS-solmussa. Vastaavasti linja-auto toisi muualta verkosta kylään osoitetut viestit mukanaan.

Kylässä sijaitsevien solmujen kannalta verkko toimii reitityksellisessä mielessä aivan kuin DTLSR-protokollaa käyttävä DTN-verkkokin. Koska yhdyskäytävä ulkomaailmaan on muuta verkkoa korkeamman tason solmu, ei se välitä koskaan reititystietoa yhteyden katkeamisesta tai nousemisesta mahdollisesti hitaaseen kyläverkkoon. Kylän asukkaiden ja WiMAX-tukiasemien kannalta yhteys on aina saatavilla yhdyskäytävänä toimivan tason 2 solmun kautta. Haittapuolena työssä esitetystä dynaamisesta reitityksestä verrattuna DTLSR-protokollaan on mahdollisesti suurempi konfigurointityö.



Kuva 20: MICS-järjestelmän soveltaminen DTLSR-protokollan esimerkitapaukseen. Numerot kuvaavat solmujen reititystasoa

### 3.6 Yhteenveto

MICS ainutlaatuisena sanomavälitysjärjestelmänä vaatii yksilöllisen tavan dynaamisen reitityksen hoitamiseen. Standardoitu linkkitilaprotokolla IS-IS antaa hyvän pohjan reititykselle, ja suhteellisen pienillä muutoksilla saadaan kokonaisuus toimimaan haastavassa ympäristössä.

Yleisesti käytössä olevat erityisvälitysverkkojen protokollat ovat MICS-verkon kannalta ongelmallisia. Stokastiset protokollat tulvittavat verkossa sanomia aiheuttaen verkkoon paljon ylimääräistä kuormaa. MICS-verkossa saattaa liikkua samanaikaisesti paljon hyötykuormaa hitaiden yhteyksien yli, jolloin stokastisen reititysmenetelmän aiheuttama ylimääräinen liikenne olisi liikaa. MANET-verkoissa käytetyt

reititysprotokollat eivät myöskään suoraan toimi, koska MICS ei takaa MANET-verkkojen vaatimaa päästä-päähän -yhteyttä.

Muitakin haasteellisissa verkoissa käytettäviä linkkitilaan perustuvia reititysprotokollia on esitetty, mutta nekin on suunniteltu varsin erilaiseen ympäristöön. DTLSR olettaa verkon olevan suhteellisen vakaa, ja linkkien katkeilun olevan joko enemmän poikkeus tai hyvin säännöllinen tapahtuma. Vaikka työssä dynaaminen reititys puolestaan on optimoitu toimimaan tietynkaltaisessa hierarkisessa verkko-topologiassa, on se kuitenkin mahdollista ottaa käyttöön myös ympäristöissä mihin DTLSR on suunniteltu.

Reititystasot ja -alueet ovat keskeinen muutos IS-IS-protokollaan verrattuna. Monitasoisella reititystasojärjestelmällä voidaan vähentää reititysliikenteen määrää mahdollisesti hitailla linkeillä verkon reunoilla, ja reititysalueilla saadaan kompaktis-  
ti pienennettyä liikenteen määrää ja tarvetta verkossa kokonaisuudessaan. Hyödyn-  
tämällä MICS-järjestelmän tarjoamia ominaisuuksia, kuten solmujen ilmoittautu-  
mista ja naapuritiedon saamista järjestelmältä, oli mahdollista kehittää optimointe-  
ja jotka mahdollistavat dynaamisen reitityksen tehokkaan toiminnan haastavissakin  
verkko-olosuhteissa.

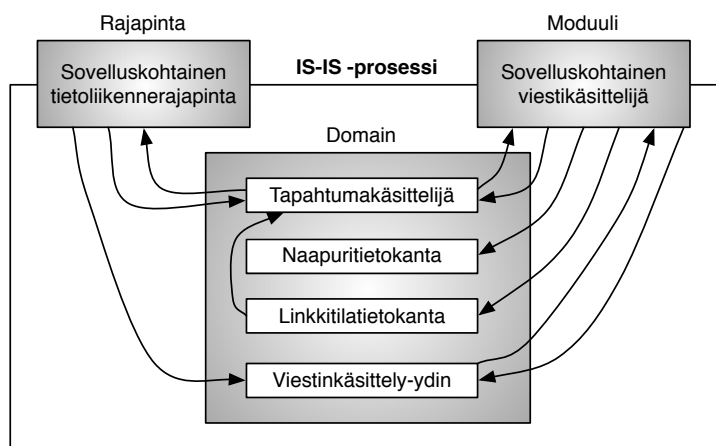
## 4 Toteutus

Tässä luvussa käydään aluksi läpi toteutetun ohjelmiston käyttämät lisäkirjastot ja esitellään sovellusarkkitehtuuri. Tämän jälkeen esitellään toteutukseen liittyvät IS-IS-protokollan sanomalaajennukset ja tutustutaan sovelluksen rajapintoihin järjestelmän muiden osien kanssa. Lopuksi esitellään käytetty testiympäristö ja käydään läpi saadut testitulokset.

### 4.1 Sovellusarkkitehtuuri

Työn yhteydessä toteutettu modulaarinen reititysydin mahdollistaa IS-IS-reitityks-protokollan käyttämisen useiden erilaisten verkkojen ja IS-IS-protokollaa hyödyn-tävien järjestelmien kanssa. Ydin käyttää hyödyksi staattisesti tai dynaamisesti la-dattavia moduuleita, joiden avulla viestin lähettäminen ja vastaanottaminen, sekä viestien luominen ja käsittely on erotettu varsinaisesta ytimestä, joka hoitaa vies-tien jäsentämisen ja sarjallistamisen, sekä tarjoaa tarvittavat tietorakenteet ja ra-japinnat yleisimmin käytössä oleviin malleihin. Kuvassa 21 esitetään sovelluksen arkkitehtuuri graafisesti. Kuvasta nähdään reititysytimen eri osien ja ladattavien moduulien väliset suhteet.

Ulkoisista paketeista toteutus hyödyntää libevent-kirjastoa [36] tapahtumapoh-jaisen ohjelmointimallin mahdollistamiseksi, sekä GLib-kirjastoa [37] tietorakentei-siin.



Kuva 21: Ohjelmiston arkkitehtuurikuvaus

Toteutus pyrittiin tekemään mahdollisimman joustavaksi ja mukautuvaksi eri tilanteisiin. Karkeasti toteutuksen osat voidaan jakaa kolmeen eri osaan: ytimeen, viestien lähettämisestä ja vastaanottamisesta vastaaviin rajapintoihin sekä viestien

käsittelystä vastaaviin moduuleihin. Rajapinnat ja moduulit ovat joko ajon aikana dynaamisesti ladattavia tai käynnistyksen yhteydessä staattisesti ladattavia komponentteja.

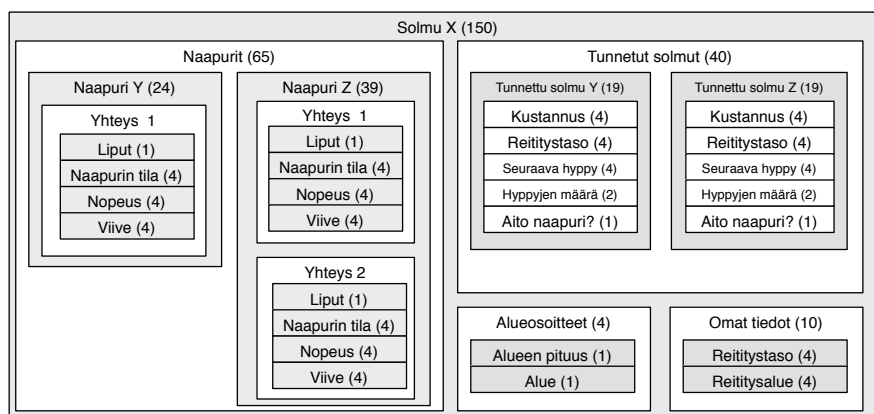
Rajapinnat (engl. interface) vastaavat kommunikoinnista tietyn viestintämedian kanssa. Yksi mahdollinen rajapinta mitä voitaisiin käyttää yleisessä toteutuksessa voisi olla esimerkiksi Ethernet-rajapinta. MICS on tässä tapauksessa yleistä tilannetta yksinkertaisempi, koska se tarjoaa omat rajapinnat eri viestintämedioille. Tästä johtuen MICS-toteutukselle riittää käyttöön yksi sähköpostia käyttävä rajapinta. Työssä toteutettu MICS-rajapinta hyödyntää sähköpostin vastaanotossa käytettävän IMAP [31] ja sähköpostin lähetyksessä käytettävän SMTP [30]-protokollien kanssa projektissa toteutettua libmailclient-kirjastoa. Varsinaisen viestin vastaanoton lisäksi rajapinnat sarjallistavat ja jäsentävät sanomat linjamuodosta sisäisiin tietorakenteisiin.

Viestien käsittelystä vastaava moduuli tarjoaa ytimen käyttöön toteutusspesifien TLV- ja ali-TLV -objektien sarjallistamiseen ja jäsentämiseen käytettävät funktiot. Ydin tarjoaa laajennuksien käyttöön mahdollisuuden rekisteröidä viestinkäsittelyfunktioit ytimen ulkopuolelta, jolloin ytimeistä erilliset moduulit pystyvät tarjoamaan toteuttamansa funktionsa ytimen käyttöön helposti. Tämän lisäksi viestinkäsittelymoduuli käsittelee tulevat sanomat ennen ydintä ja lähtevät sanomat ytimen jälkeen, jolloin ydin voidaan pitää toteutuksesta ja käyttöympäristöstä riippumattomana. Vastaavasti moduulit hoitavat toteutuksen vaatimat lisäykset standardin IS-IS-toiminnallisuuden lisäksi. Viestinkäsittelymoduuli hoitaa kaikkien viestien kokoamisen, joko ytimeltä saapuvan tai moduulin sisäisen signaalin perusteella. MICS-toteutuksessa tämä moduuli hoitaa myös SPF-laskennan suorittamisen ja laskentatuloksen edelleenvälittämisen D-Bus-väylän ylitse reitittimelle.

## 4.2 MICS-spesifiset lisäykset

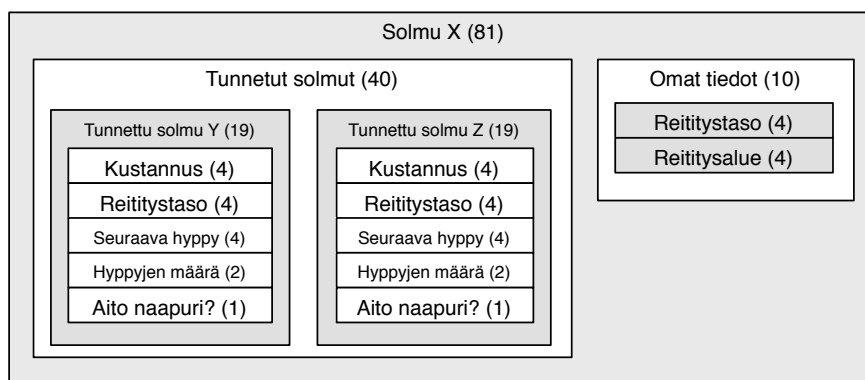
Jotta IS-IS-reititysydin saatiin toimimaan halutulla tavalla MICS-ympäristössä, täytyi sovelluksen MICS-spesifisten komponenttien kautta tehdä laajennuksia ja muutoksia protokollastandardiin. Tässä luvussa käydään läpi tehdyt muutokset.

Kuvassa 22 on esitetty esimerkki MICS-järjestelmän yhteydessä käytettävistä linkkitilaviesteistä. Jokainen linkkitilaviesti sisältää sekä tietoa sen lähettäneestä solmusta, listan kaikista solmun naapureista ja eri yhteyksistä naapureihin, että listauksen kaikista solmun tietämistä MICS-solmuista ja niiden reititystasoista. Tässä esimerkissä linkkitilaviestin lähettäneellä solmulla X on kaksi naapuria, Y ja Z. X:n ja Y:n välillä on yksi yhteys, ja X:n ja Z:n välillä kaksi yhteyttä.



Kuva 22: Linkkitilaviestien sisältö

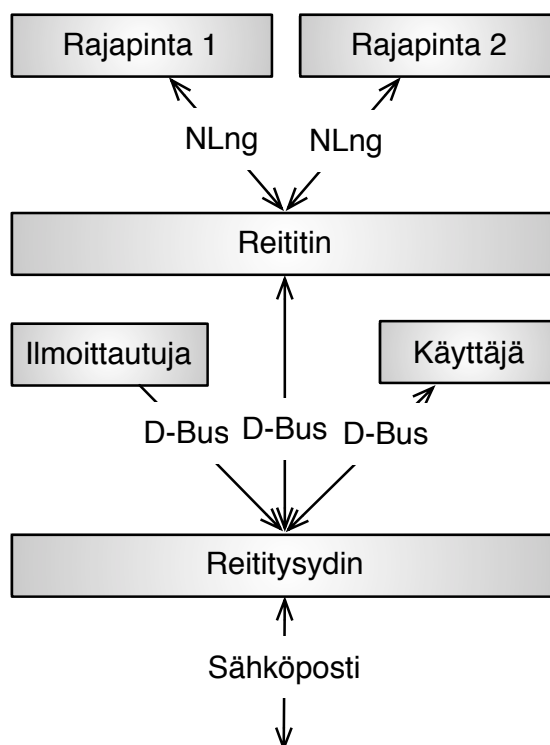
Kuvassa 23 esitetään reititusalueiden välillä kulkevien reititusalueviestien rakenne. Tämä viestityyppi on täysin MICS-spesifinen laajennus. Reititusalueiden reunasolmut lähettävät toisilleen säännöllisesti tiedon solmuista joille tietävät reitin, jolloin reititys eri alueiden välillä on mahdollista maltillisella kaistankäytöllä. Seuraava hyppy -kenttää käytetään suodattamaan pois naapurin kertomat solmut, jotka naapuri on oppinut meidän kautta. Jos solmu C on saavutettavissa vain solmun B kautta, ja B kertoo tämän A:lle, niin ei ole järkeä että B yrittäisi reitittää C:lle A:n kautta vaikka A kertoisikin C:n olevan saavutettavissa kauttaan. Koska A osaa kertoa C:n olevan saavutettavissa kauttaan solmun B kautta, ei B ota C:tä huomioon A:n kautta saavutettavana solmuna. Hyppyjen määrä -kenttää käytetään taas mahdollisten reitityskierteiden katkaisemiseen. Jos solmu kertoo toisen solmun olevan kauttaan saavutettavissa liian monen hypyn kautta, oletetaan verkossa olevan reitityskierteen eikä kyseistä solmua oteta huomioon reitinlaskennassa. Näin saadaan kierre katkaistua ja olematon solmu hävitettyä verkosta.



Kuva 23: Reititusalueviestien sisältö

## Kommunikointi prosessin ulkopuolelle

Reititysydin käyttää kahta väylää kommunikointiin muiden prosessien kanssa. D-Bus-väylän yli vaihdetaan tietoa paikallisen MICS-solmun kanssa, ja sähköposti-rajapintaa käytetään kommunikointiin muilla MICS-solmuilla ajettavien reititysydin-prosessien kanssa. Kuva 24 esittää rajapinnat ja komponentit jotka vaikuttavat reititysytimen toimintaan.



Kuva 24: Reititysytimen rajapinnat

D-Bus on sovelluksien väliseen kommunikointiin (Inter-Process Communication, IPC) tarkoitettu julkaisija-tilaaja (Publish - Subscribe) -mallin signaaleilla toteuttava viestiväylä. Viestintä tapahtuu käyttämällä joko metodikutsuja ("kerro tilasi") tai kuuntelemalla ja reagoimalla eri tapahtumia ilmaisevia signaaleja. D-Bus on erityisesti Linux-maailmassa yleisesti käytetty kommunikointiväylä sovelluksien välillä.

D-Bus-väylän yli kuunnellaan seuraavia tapahtumia:

- Viestintärajapintojen lisäys ja poisto
- Naapurin tilan muutos (sisältää uudet naapurit)
- Naapurin poistuminen

- Viestintärajäpinnan tilan muutos
- Reittien päivitys (sisältää uudet reitit)
- Reittien poisto
- Ilmoittautumisen lisäys
- Ilmoittautumisen poisto

Näiden tapahtumien kuuntelun lisäksi D-Bus-väylän yli pyydetään sovelluksen käynnistymisen yhteydessä tämän hetkinen tilanne. Kun alkutilanne on selvillä, pysyy sovelluksen sisäinen tila ajantasalla seuraamalla järjestelmältä tulevia muutos-signaaleita.

### 4.3 Testaus

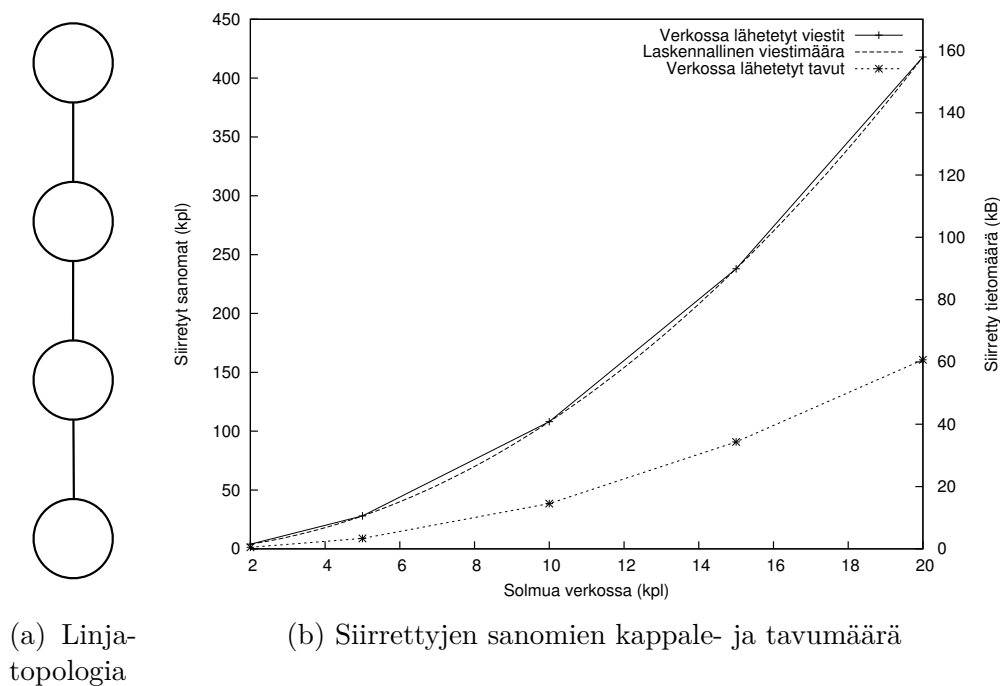
Ohjelmiston testaus ja mittaukset tätä työtä varten suoritettiin simuloidussa verkossa yhden isäntäkoneen sisällä. Jokainen solmu ajettiin eristetyksi omassa erittäin kevyessä virtuaali-Linuxissa (Linux Container, LXC), ja solmujen välinen liikenne toteutettiin tietoverkkosimulaattori NS3:n avulla.

Testeissä tutkittiin sovelluksen verkkoon aiheuttamaa kuormaa sekä sanomaa että tavumäärällisesti erilaisissa verkkotopologioissa ja erilaisilla konfiguraatioilla. Sanomamäärä verkossa tarkoittaa kaikkien solmujen yhteensä generoimaa sanomamäärää, ja vastaavasti tavumäärä tarkoittaa kaikkien solmujen lähettämää tavumäärää verkkoon, lukuunottamatta IP-verkkorajapinnan jokaiseen viestiin lisäämää pientä otsaketta. Tavumäärä ei siis ole reititysytimen eteenpäin lähettämä tavumäärä, vaan viestikäsittelijältä pakkauksen, salauksen ja natiivimuotoon muuttamisen jälkeen lähtevä tavumäärä.

#### Linjatopologiaverkon muodostuminen

Ensimmäisessä testitapauksessa käytiin läpi kuvan 25a esittämän linjatopologian mukaisten erikokoisten verkkojen toimintaa. Kyseisessä topologiassa verkon solmut muodostavat yhtenäisen pitkän linjan, jossa linjan päätysolmuilla on yksi naapuri ja kaikilla muilla verkon solmuilla kaksi naapuria.

Testeissä saadut mittaustulokset esitetään kuvassa 25b. Kuvasta nähdään, että toteutus vastaa täysin laskennallisia viestimääriä näille verkoille.



Kuva 25: Testauksessa käytetty linjatopologia ja testitulokset

### Täysin kytketyn verkon muodostuminen

Toisena verkkotopologiana testattiin sovelluksen toimintaa kuvan 26a esittämässä täysin kytketyssä verkossa. Täysin kytketty verkkotopologia on reitityksen kannalta linjatopologiaan verrattuna paljon raskaampi: verkkokuvan muodostuessa jokainen solmu lähettää sanomia jokaiselle toiselle, kun linjatopologian tapauksessa solmulla on korkeintaan kaksi naapuria joille sanoa pitää välittää.

Täysin kytketyn verkon mittaustulokset esitetään kuvassa 26b. Verrattuna linjatopologiaan huomataan samankokoisessa verkossa lähetettävän viestejä ja tavuja huomattavasti enemmän.

### Kahden täysin kytketyn verkkoalueen yhdistyminen

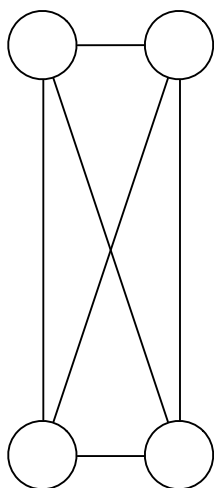
Kahden täysin kytketyn verkkoalueen yhdistymisessä tutkittiin tapausta, jossa verkkoalueet ovat itse stabiilissa tilassa reitityksen suhteen ennen yhdistymistä. Kuvassa 27a on esitetty tilanne yhdistymisen jälkeen.

Mittauksista saadut tulokset vastaavat laskennallisia arvoja.

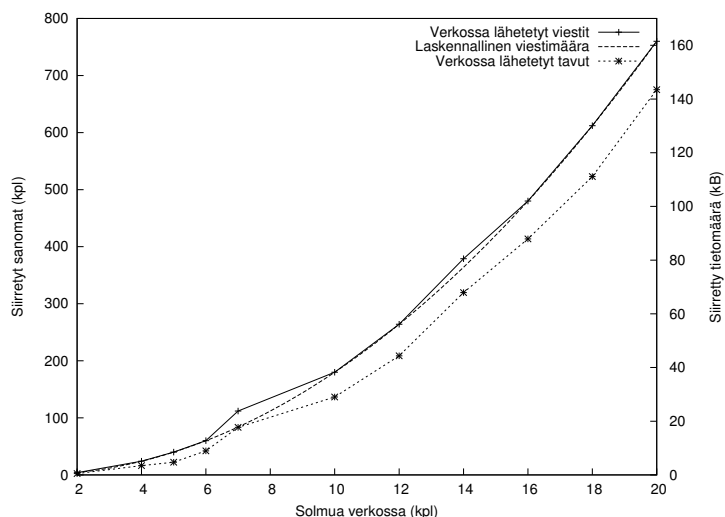
### Kolmen verkkoalueen välinen reititys erilaisilla konfiguraatioilla

Tässä testissä tutkittiin sovelluksen toimintaa kuvan 28a mukaisessa verkossa erikokoisilla verkkoalueilla ja erilaisilla konfiguraatioilla. Testeissä käytettiin yhteensä



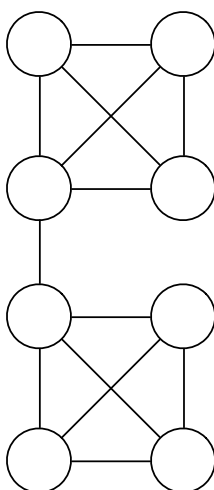


(a) Täysin kytketty verkkotopologia

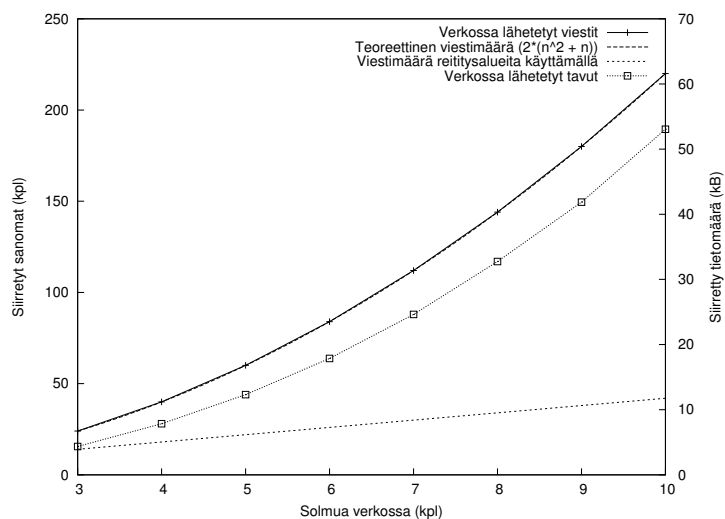


(b) Siirrettyjen sanomien kappale- ja tavumäärä

Kuva 26: Testauksessa käytetty täysin kytketty verkkotopologia ja testitulokset



(a) Verkkoalueiden yhdistäminen

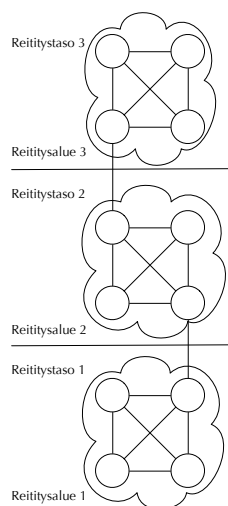


(b) Siirrettyjen sanomien kappale- ja tavumäärä

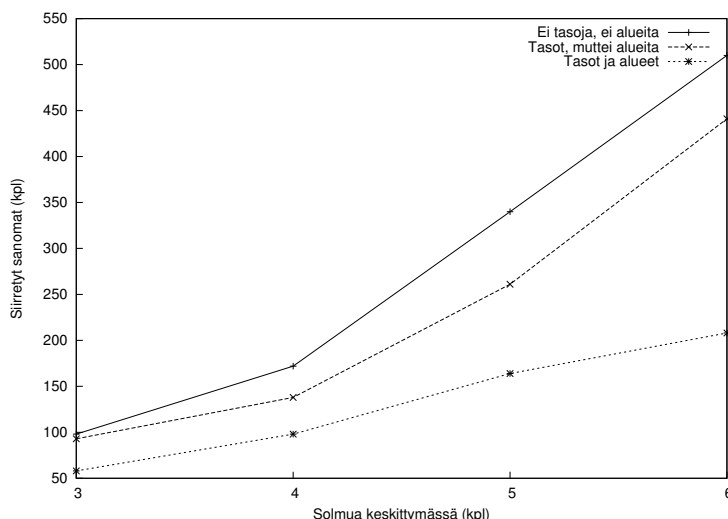
Kuva 27: Testauksessa käytetty verkkotopologia ja testitulokset

kolmea erilaista konfiguraatiota:

- Ryppäät ovat samalla reititystasolla ja -alueella
- Ryppäät ovat samalla reititysalueella, mutta eri reititystasoilla
- Ryppäät ovat sekä eri reititysalueilla että tasoilla



(a) Verkkoalueiden yhdistäminen, alueet mukana



(b) Siirrettyjen sanomien kappale- ja tavumäärä

Kuva 28: Testauksessa käytetty verkkotopologia ja testitulokset

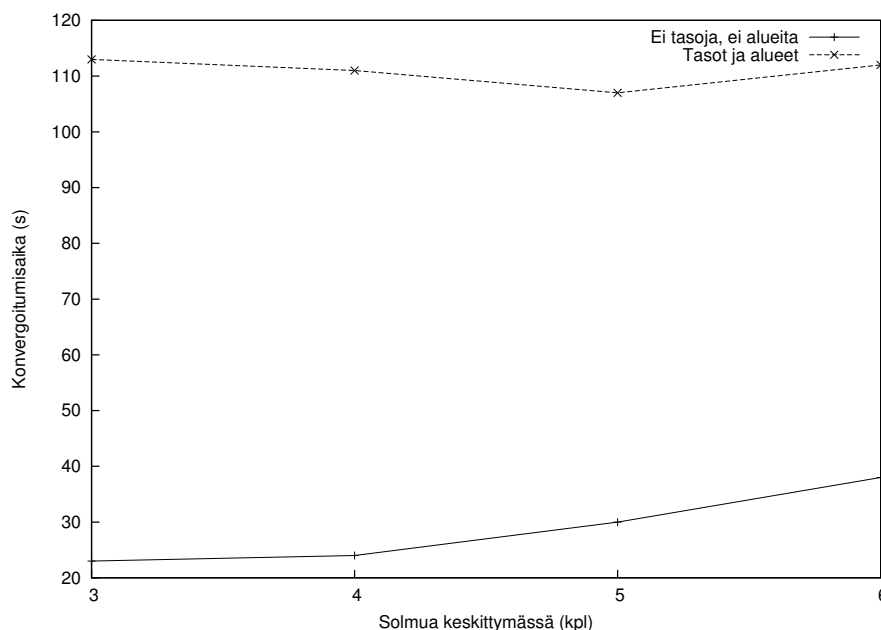
Testin tuloksista nähdään hyvin reititysalueiden vaikutus reititykseen: siirrettävä viestimäärä tippuu huomattavasti, mutta samalla verkon konvergoitumisaika kasvaa huomattavasti (kuva 29). Tämän perusteella onkin suositeltavaa, että solmut joille kommunikoidaan todennäköisemmin löytyvät saman reititysalueen sisältä jotta verkossa tapahtuviin muutoksiin voidaan reagoida nopeasti.

#### 4.4 Kehittämiskohteet

Kuten testituloksista nähdään, toimii työssä toteutettu reititysydin monessa tapauksessa halutulla tavalla. Tästä huolimatta varsinkin ohjelmiston sisäistä toteutusta on mahdollista parantaa monella eri tavalla.

Tärkein yksittäinen kehittämiskohde työssä toteutetulle ohjelmistolle on runsas testaaminen ja pullonkaulatapauksien hakeminen. Vaikka sovellusta testattiinkin työn aikana sekä simuloiduissa että oikeissa verkoissa hyvällä menestyksellä, on varmasti monia tapauksia missä sovellus ei toimi halutulla tavalla. Koska järjestelmä kehittyi ja halutut ominaisuudet elivät jatkuvasti kehitystyön aikana, on varmasti olemassa tapauksia missä kehitetty malli ei toimi tämän hetkisen suunnittelun pohjalta. Myöskin virheet ohjelmistoissa ovat hyvin yleisiä, joten testaaminen monella eri tasolla on tärkeää sovelluksen kriittisen luonteen vuoksi. Testeihin liittyen myös yksikkötestien lisääminen ohjelmiston laadun parantamiseksi olisi suositeltavaa.

Toinen selvä kehityskohde olisi hyödyntää sähköpostirajapinnan mahdollista-



Kuva 29: Verkon konvergoitumisaika kolmen täysin kytketyn verkkoalueen tapauksessa

maa monilähetystä. Koska sovellukselle jokainen naapuri näkyy point-to-point -tyyppisenä linkkinä, käsittelee sovellus myöskin jokaisen naapurin erillisenä linkkinä. Kuitenkin esimerkiksi omaa linkkitilatietaa lähetettäessä olisi mahdollista luoda sanoma vain kerran, merkitä kaikki halutut vastaanottajat sanomaan, ja lähettää viesti järjestelmälle eteenpäin vain kerran. Tämä pienentäisi ohjelmiston aiheuttamaa prosessori- ja siirräntäkuormaa huomattavan paljon.

Kolmantena asiana sovelluksen käyttämät sanomat eivät ole tällä hetkellä optimoituja koon suhteen, vaan niissä lähetetään mukana tietoa jota tällä hetkellä ei hyödynnetä lainkaan. Vaikka Viestikäsittelijä pakkaakin lähetettävät sanomat, olisi silti hitaissa verkoissa tärkeää optimoida MICS-ytimen komponenttien käyttämiä sanomakokoja, jotta hyötykuormalle jäisi verkossa mahdollisemman paljon tilaa.

Vaikka työn alussa yhdeksi kriteeriksi määriteltiin tehokkuus suorittimen ja muistinkäytön suhteen, olisi toteutetussa sovelluksessa niihin liittyen optimoimisen varaa. Toistaiseksi sovellusta on ajettu suhteellisen tehokkaissa laitteistoissa, joten optimoiminen ei ole ollut yhtä kriittistä kuin virheettömän toiminnallisuuden saavuttaminen. Tästä huolimatta, varsinkin pienitehoisten sulautettujen laitteiden lisääntyessä maailmalla jatkuvasti, olisi tähänkin puoleen panostettava huolellisemmin.

## 4.5 Yhteenveto

Tässä luvussa esiteltiin ohjelmiston sisäistä arkkitehtuuria ja toteutuksessa käytettyjä IS-IS-laajennuksia. Lisäksi käytiin läpi sovellukselle suoritettuja testejä ja mahdollisia parannuskohteita.

Testituloksista nähdään, että toteutettu sovellus toimii odotetulla tavalla ja suoriutuu reitityksestä kohtuullisilla sanoma- ja tiedonsiirtomäärillä. Kuitenkin parantamisen varaa on:

- Kattavampaa ohjelmistotestausta sovelluksen kriittisen luonteen vuoksi
- Sähköpostirajapinnan tarjoaman monilähetyksen hyödyntäminen
- Sanomien koot eivät ole optimoituja, vaan niissä lähetetään melko paljon ylimääräistä tietoa
- Tietorakenteiden ja algoritmien läpikäyminen ja optimointi pienitehoisten laitteiden yleistyessä

Kokonaisuutena voidaan todeta, että toteutettu ohjelmisto toimii sille asetettujen kriteerien mukaisesti, mutta käyttöympäristöjen ja -vaatimusten lisääntyessä lisäkehitystä tarvitaan.

## 5 Yhteenveto

Tässä työssä käytiin läpi reitityksen perusteita erilaisissa verkoissa ja esiteltiin toteutus dynaamisesta reitityksestä haasteelliseen sanomanvälitys verkkoon. Työ koostui ohjelmiston suunnittelusta, määrittelystä, toteutuksesta, testauksesta ja dokumentoinnista.

Työ rakentui IS-IS -reititysprotokollan ja MICS-sanomanvälitysohjelmiston päälle. IS-IS on yleisesti käytössä oleva ja tunnettu linkkitilareititysprotokolla, jonka vahvuuksia on protokollariippumattomuus ja helppo laajennettavuus. MICS on haasteellisiin ympäristöihin suunniteltu sanomanvälitysjärjestelmä, joka yhdistää erilaisia verkkotekniikoita keskenään muodostamalla fyysisen verkkotopologian päälle oman päällysverkkonsa. Tässä työssä toteutettiin IS-IS -protokollaan pohjautuva reititysratkaisu MICS-päällysverkolle.

MICS on alustana haasteellinen dynaamiselle reititykselle heterogeenisestä luonteesta johtuen. Yhdessä verkossa keskenään reititettävien solmujen välillä voi olla sekä erittäin hitaita, että erittäin nopeita linkkejä. Kuitenkin reitityksen pitää pystyä reagoimaan verkossa tapahtuviin muutoksiin nopeasti ilman, että verkossa sijaitsevat hitaat linkit ylikuormittuvat. Testaustuloksista nähdään, että toteutus saatiin toimimaan MICS-verkossa halutulla tavalla hyödyntämällä alustan tarjoamia ominaisuuksia kuten ajantasaista naapuritietoa, että IS-IS -protokollaan tehtyjä laajennuksia: tulvittamisen rajoittamista, n-tasoista reititystä sisältäen erityisen 0-reititystason, ja reititysalueita.

Muihin haasteellisiin verkkoihin tehtyihin linkkitilaprotokolliin verrattuna työssä lähdettiin liikkeelle erilaisista lähtökohdista. Delay Tolerant Link-State Routing on suunniteltu suhteellisen vakaisiin verkkoihin, joiden yhteys ulkomaailmaan katkeaa silloin tällöin. Tässä opinnäytetyössä taas optimoitiin reitityksen suorituskykyä topologiaaltaan hierarkiseen verkkoon, jossa hitaimmat yhteydet sijaitsevat aivan verkon reunalla. Aikaisemmissa haasteellisten verkkojen linkkitilaprotokollissa on siis ratkaistu toisenlaista ongelmaa kuin tässä työssä. Tästä huolimatta voitiin esittää työssä toteutettu reititysmalli toimivaksi myös ympäristössä mihin aikaisemmin toteutettu DTN-verkkojen linkkitilaprotokolla DTLSR on suunniteltu, mikä kertoo suunnitellun mallin joustavuudesta eri ympäristöihin.

Vaikka työssä suunniteltu reititys toimiikin testeissä halutulla tavalla, on toteutuksessa vielä optimoimisen varaa erityisesti sanomien lähetyksen, sanomakoon ja prosessorikäytön suhteen. Käytettyjä sisäisiä tietorakenteita on mahdollista optimoida tiettyihin tilanteisiin sopivammiksi, ja myös MICS-järjestelmän sähköpostirajapinnan käyttöä on mahdollista tehostaa: esimerkiksi omaa linkkitilaa levittäessä on turhaa generoida jokaiselle naapurille omaa sanomaa; sen sijaan kannattaisi yhteen

sanomaan laittaa monta vastaanottajaa ja antaa järjestelmän hoitaa levitys jokaiselle naapurille parhaaksi katsomallaan tavalla, esimerkiksi mahdollisesti hyödyntäen monilähetysverkkoa.

Johdannossa työn tavoitteiksi valittiin luotettavuus, skaalautuvuus, tehokkuus, pieni kaistakäyttö ja laajennettavuus. Skaalautuvuus, tehokkuus ja pieni kaistankäyttö ovat toteutuneet reitityksen kanssa tehtyjen optimointejen kanssa vähintään kohtuullisesti, vaikka optimointia varsinkin tehokkuuden ja pienen kaistankäytön suhteen on mahdollista tehdä. Käyttötapauksesta riippuen voidaan parantaa tehokkuutta asettamalla kaikki solmut samalla reititysalueelle ja korkealle reititystasolle, tai skaalautuvuutta ja pientä kaistankäyttöä lisäämällä reititystasoja ja alueita verkkoon. Toteutus on ehdottomasti helposti laajennettavissa standardin IS-IS-protokollan tapaan uusilla TLV-tyypeillä ja viestikäsittelijöillä tarvitaessa. Toteuttu ohjelmisto on myös testeissä toiminut luotettavasti, iso osa oudoista virhetilanteista testien aikana on johtunut virheellisestä verkkokonfiguraatioista. Tämän perusteella voidaan sanoa, että työ täyttää sille asetetut tavoitteet.

Tämän opinnäytetyön tulokset on julkaistu The 2nd IEEE World Symposium on Web Applications and Networking 2015 -konferenssissa Tunisiassa maaliskuussa 2015 [38].

## Viitteet

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844, 9780262033848.
- [2] R. Bellman. On a Routing Problem. *Quarterly of Applied Mathematics*, 16: 87–90, 1958.
- [3] L. R. Ford. Network Flow Theory. Report P-923, The Rand Corporation, 1956.
- [4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. URL <http://gdzdoc.sub.uni-goettingen.de/sub/digbib/loader?did=D196313>.
- [5] G. Malkin. RIP version 2. Internet Engineering Task Force, Request for Comments 2453, November 1998. URL <https://tools.ietf.org/rfc/rfc2453.txt>.
- [6] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). Internet Engineering Task Force, Request for Comments 4271, January 2006. URL [www.ietf.org/rfc/rfc4271.txt](http://www.ietf.org/rfc/rfc4271.txt).
- [7] ISO/IEC 10589. Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473), 2002.
- [8] J. Moy. OSPF version 2. Internet Engineering Task Force, Request for Comments 2328, April 1998. URL <https://www.ietf.org/rfc/rfc2328.txt>.
- [9] International telecommunication union. key 2006-2013 ict data for the world. <http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>. Haettu: 21.3.2014.
- [10] J. Postel. Internet protocol. Internet Engineering Task Force, Request for Comments 791, September 1981. URL <http://tools.ietf.org/html/rfc791>.
- [11] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. Internet Engineering Task Force, Request for Comments 4632, August 2006. URL <http://tools.ietf.org/html/rfc4632>.

- [12] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss. Interplanetary Internet (IPN): Architectural Definition. Internet Engineering Task Force, Internet Draft, May 2001.
- [13] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 27–34. ACM, New York, NY, USA, 2003. ISBN 1-58113-735-4. URL <http://doi.acm.org/10.1145/863955.863960>.
- [14] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. Internet Engineering Task Force, Request for Comments 4838, April 2007. URL <http://www.ietf.org/rfc/rfc4838.txt>.
- [15] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University Department of Computer Science, July 2000.
- [16] R. Ramanathan, P. Basu, and R. Krishnan. Towards a formalism for routing in challenged networks. In *Proceedings of the Second ACM Workshop on Challenged Networks*, CHANTS '07, pages 3–10. ACM, New York, NY, USA, 2007. ISBN 978-1-59593-737-7. URL <http://doi.acm.org/10.1145/1287791.1287794>.
- [17] A.V. Vasilakos, Y. Zhang, and T. Spyropoulos. *Delay Tolerant Networks: Protocols and Applications*. Wireless Networks and Mobile Communications. CRC Press, 2012. ISBN 9781466513013. URL <http://books.google.fi/books?id=2ERN5lgs3AwC>.
- [18] A. Jamalipour and Y. Ma. *Intermittently Connected Mobile Ad Hoc Networks - from Routing to Content Distribution*. Springer Briefs in Computer Science. Springer, 2011. ISBN 978-1-4614-1572-5. i-xviii, 1-107 pp. URL <http://dx.doi.org/10.1007/978-1-4614-1572-5>.
- [19] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, pages 90–. IEEE Computer Society, Washington, DC, USA, 1999. ISBN 0-7695-0025-0. URL <http://dl.acm.org/citation.cfm?id=520551.837511>.



- [20] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). Internet Engineering Task Force, Request for Comments 3626, October 2003. URL <http://www.ietf.org/rfc/rfc3626.txt>.
- [21] Z.J. Haas, M.R. Pearlman, and P. Samar. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. Internet Engineering Task Force, Internet Draft, July 2002. URL <http://www.ietf.org/archive/id/draft-ietf-manet-zone-zrp-04.txt>.
- [22] Z.J. Haas, M.R. Pearlman, and P. Samar. The Intrazone Routing Protocol (IARP) for Ad Hoc Networks. Internet Engineering Task Force, Internet Draft, July 2002. URL <https://tools.ietf.org/html/draft-ietf-manet-zone-iarp-02>.
- [23] Z.J. Haas, M.R. Pearlman, and P. Samar. The Interzone Routing Protocol (IERP) for Ad Hoc Networks. Internet Engineering Task Force, Internet Draft, July 2002. URL <https://tools.ietf.org/html/draft-ietf-manet-zone-ierp-02>.
- [24] M. Demmer and K. Fall. DTLSR: delay tolerant routing for developing regions. In *NSDR '07: Proceedings of the 2007 workshop on Networked systems for developing regions*, pages 1–6. ACM, New York, NY, USA, 2007. ISBN 978-1-59593-787-2. URL <http://www.cs.berkeley.edu/~kfall/papers/dtlsr-nsdr07.pdf>.
- [25] Yuanyuan M., Yang X., Z. Bong, Chai K. Y., A.Y.P. Chua, A.J.L. Tan, A. Sivagami, Tiah K. N., Kok S. L., and Ser H. T. Multi-policy link state routing for disruption tolerant networks. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pages 1–7, April 2013.
- [26] R. Järvinen, J. Määttä, R. Luostarinen, J. Manner, and M. Luoma. Mics messaging platform: Architecture, design and routing. In *Military Communications Conference, 2010 - MILCOM 2010*, pages 1893–1898, Oct 2010. ISSN 2155-7578.
- [27] D-bus home page. <http://www.freedesktop.org/wiki/Software/dbus/>. Haettu: 23.4.2015.
- [28] J. Määttä, R. Järvinen, R. Luostarinen, and J. Manner. The virtual network system. In *Proceedings of the 2Nd International Workshop on Middleware for Pervasive Mobile and Embedded Computing, M-MPAC '10*, pages 3:1–3:8. ACM,

- New York, NY, USA, 2010. ISBN 978-1-4503-0451-1. URL <http://doi.acm.org/10.1145/1890791.1890794>.
- [29] J. Salim, H. Khosravi, A. Kleen, and A. Kuznetsov. Linux Netlink as an IP Services Protocol. Internet Engineering Task Force, Request for Comments 3549, July 2003. URL <http://www.ietf.org/rfc/rfc3549.txt>.
  - [30] J. Klensin. Simple Mail Transfer Protocol. Internet Engineering Task Force, Request for Comments 5321, October 2008. URL <http://www.ietf.org/rfc/rfc5321.txt>.
  - [31] M. Crispin. Internet Message Access Protocol - Version 4rev1. Internet Engineering Task Force, Request for Comments 3501, March 2003. URL <http://www.ietf.org/rfc/rfc3501.txt>. Updated by RFCs 4466, 4469, 4551, 5032, 5182, 5738, 6186, 6858.
  - [32] H. Gredler and W. Goralski. *The Complete IS-IS Routing Protocol*. Springer London, 2005. ISBN 9781846281051.
  - [33] D. Katz and D. Ward. Bidirectional Forwarding Detection (BFD). Internet Engineering Task Force, Request for Comments 5880, June 2010. URL <http://www.ietf.org/rfc/rfc5880.txt>. Updated by RFC 7419.
  - [34] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Internet Engineering Task Force, Request for Comments 3031, January 2001. URL <http://www.ietf.org/rfc/rfc3031.txt>.
  - [35] R. Aggarwal, K. Kompella, T. Nadeau, and G. Swallow. Bidirectional forwarding detection (bfd) for mpls label switched paths (lsp). RFC 5884, RFC Editor, June 2010. URL <http://www.rfc-editor.org/rfc/rfc5884.txt>. <http://www.rfc-editor.org/rfc/rfc5884.txt>.
  - [36] libevent: an event notification library. <http://libevent.org>. Haettu: 2015-04-06.
  - [37] Glib reference manual. <https://developer.gnome.org/glib/>. Haettu: 2015-04-06.
  - [38] R. Järvinen, A. Jaakkola, J. Määttä, L. Liuhto, R. Luostarinen, J. Manner, and M. Luoma. Hierarchical link-state routing in disruption-tolerant networks. In *The 2nd IEEE World Symposium on Web Applications and Networking 2015*, 2015.